Engineering and Technology Journal e-ISSN: 2456-3358

Volume 11 Issue 06 June-2026, Page No.- 10496- 10504

DOI: 10.47191/etj/v11i06.25, I.F. – 8.595

Big Data Analytics Algorithms for Analyzing Employee Leave Patterns in Semarang City using Apache Spark

Rara Sriartati Redjeki[1], Eko Nur Wahyudi[2], Endang Lestariningsih[3], Eka Ardhianto[4], Subkhan Indra Gunawan[5]

[1]Program Study of Information Systems, Faculty of Information Technology and Industry, Universitas Stikubank, Semarang, Central Java, Indonesia

[2]Program Study of Informatics Enginering, Faculty of Information Technology and Industry, Universitas Stikubank, Semarang, Central Java, Indonesia

[3]Program Study of Hospitality, Faculty of Vocational, Universitas Stikubank, Semarang, Central Java, Indonesia

[4,5]Program Study of Information Technology, Faculty of Information Technology and Industry, Universitas Stikubank, Semarang, Central Java, Indonesia

ABSTRACT: Leave is an employee right that plays a vital role in maintaining the balance between work productivity and human resource well-being in the workplace. Leave application patterns formed over time can reflect workload dynamics, organizational unit characteristics, and the effectiveness of operational planning. However, in organizations with large numbers of employees and high data volumes, analyzing leave application patterns is often suboptimal when relying on conventional relational database approaches. This study aims to analyze employee leave application patterns in Semarang City by applying a Big Data Analytics approach based on Apache Spark. The dataset used consists of over 150,000 leave application records from the 2020–2025 period, encompassing information on application timing, organizational units, and leave duration. The analysis process was conducted through extraction, transformation, and loading (ETL) stages using Apache Spark DataFrames, including transforming leave data into a daily basis and temporal aggregation. Furthermore, the K-Means algorithm was used to cluster leave application behavior patterns, while FP-Growth was applied to identify frequently occurring combinations of leave timing. The results indicate that most leave applications were made on weekdays with short durations; however, specific clusters showed a tendency to take leave adjacent to or sandwiching public holidays and weekends. These findings demonstrate the existence of recurring and structured leave behavior patterns, which can be utilized as a basis for evaluation and formulation of more effective and adaptive employee leave management policies.

KEYWORDS: Apache Spark; Big Data Analytics; Employee Leave; FP-Growth; K-Means Clustering.

I. INTRODUCTION

Employee leave is a constitutional right granted to workers to take time off or vacations without reducing their salary or wages (Hadad, 2024). In practice, employee leave management is a crucial pillar of human resource management (HRM) in government agencies, as it directly impacts the continuity of public services and the well-being of the employees themselves. In institutions with a large workforce, such as the Semarang City Government, leave applications occur massively, generating a substantial cumulative dataset characterized by high volume and dynamic velocity over time. The resulting patterns reflect not only individual employee needs but also serve as key indicators of workload dynamics, organizational unit characteristics, and the effectiveness of organizational operational planning.

However, in reality, human resource data management and analysis in many government agencies still rely on conventional Relational Database Management Systems (RDBMS) and administrative-oriented static reporting. While this traditional approach is sufficient for routine daily operations, it presents significant technical limitations when dealing with large-scale data (big data) stored over long historical periods, which requires complex temporal pattern analysis. A number of studies emphasize that a big data analytics approach is highly necessary to enhance data analysis capabilities in the public sector, particularly in extracting hidden patterns, trends, and temporal correlations that are impossible to identify through conventional systems (Chen & Storey, 2012).

Recent developments in Big Data Analytics technology have driven advancements in data analysis capabilities, making them more efficient and comprehensive through distributed computing architectures. This distributed computing framework is capable of handling the exponentially increasing volume of administrative data.

Modern studies show that utilizing big data analytics in the public sector not only increases data processing efficiency but also strengthens time-series and pattern-based analysis to support data-driven or evidence-based policymaking (Mergel et al., 2019).

One of the most reliable and widely used big data frameworks is Apache Spark, designed to support parallel distributed data processing using an in-memory computing approach. This architecture allows Spark to process large-scale data much faster compared to conventional disk-based SQL approaches that rely on repeated read/write cycles (Zaharia et al., 2016). The effectiveness of Apache Spark in handling large-scale data processing has been proven through various empirical studies across diverse complex analytical scenarios (Armbrust et al., 2020). In the context of temporal analysis, the application of techniques such as temporal aggregation, clustering, and frequent pattern mining on top of the Spark platform has proven effective in uncovering systematic behavioral trends (Armbrust et al., 2020; Zaharia et al., 2016).

Nonetheless, research that specifically integrates clustering methods (such as K-Means) and frequent pattern mining (such as FP-Growth) to examine employee leave application patterns in the public sector remains highly limited. One crucial aspect that has not been deeply explored is the correlation between employee leave applications and the calendar of national holidays or collective leaves. This phenomenon often triggers significant surges in leave applications (known locally as the "sandwich day"), which potentially poses a risk of service disruption or a decline in the quality of public services if not managed properly.

Based on this research gap, this study applies an Apache Spark-based Big Data Analytics approach to analyze employee leave application patterns in Semarang City, utilizing historical data from the 2020–2025 period comprising more than 150,000 records. This study focuses on descriptive and exploratory analysis, including the formation of leave behavior clusters using the K-Means algorithm, as well as the identification of leave timing and type combination patterns using the FP-Growth algorithm. The primary contribution of this research is to provide empirical insights and data-driven scientific recommendations that can be utilized by policymakers for evaluation, planning, and formulating more effective, adaptive, and accountable employee leave management policies within the Semarang City Government environment.

II. METHOD

Research Approach and Stages

The approach employed in this study is a descriptive-exploratory quantitative approach utilizing Big Data Analytics. Exploratory research aims to enhance understanding and discover new insights regarding specific phenomena, describe systemic occurrences, and explain the underlying processes of a phenomenon to define problems more granularly or develop hypotheses without direct testing. Furthermore, exploratory research formulates more relevant research questions so that subsequent studies can provide answers to future institutional challenges (Mudjiyanto, 2018). This approach was selected because the research focuses on processing and analyzing large-scale numerical data to objectively identify leave application patterns and trends, without performing hypothesis testing or predictive modeling. The descriptive-exploratory approach allows the researcher to excavate data characteristics and hidden patterns that emerge naturally from historical employee leave data.

Large-scale data processing is executed using the Apache Spark framework. This framework offers a comparative advantage in big data computing by implementing an in-memory processing mechanism, which significantly reduces dependency on disk-based operations, thereby accelerating computational execution time (Utami & Astuti, 2024). The utilization of big data has proven crucial in supporting comprehensive data-driven decision-making (Oktavia et al., 2024), effectively overcoming the limitations of conventional or manual methods that are heavily restricted by sample sizes and sampling timeframes. The entire analytical process in this study is integrated through the Extract, Transform, and Load (ETL) stages to guarantee data quality, validity, and consistency prior to advanced analysis.

The analytical phases in this study encompass three main pillars:

- Temporal Aggregation: Applied to identify the distribution and fluctuations of leave applications based on time dimensions.
- Clustering: Utilizing the K-Means algorithm to group the characteristics and behavioral patterns of employee leave applications into distinct segments.
- Frequent Pattern Mining: Applying the FP-Growth algorithm to identify frequently occurring combinations of leave timing (frequent itemsets).

The combination of these analytical techniques is leveraged to obtain a comprehensive overview of employee leave patterns based on temporal dimensions and duration characteristics. To provide a systematic overview, the complete stages of this study are visually mapped into the research flowchart presented in Figure 3.1 below.

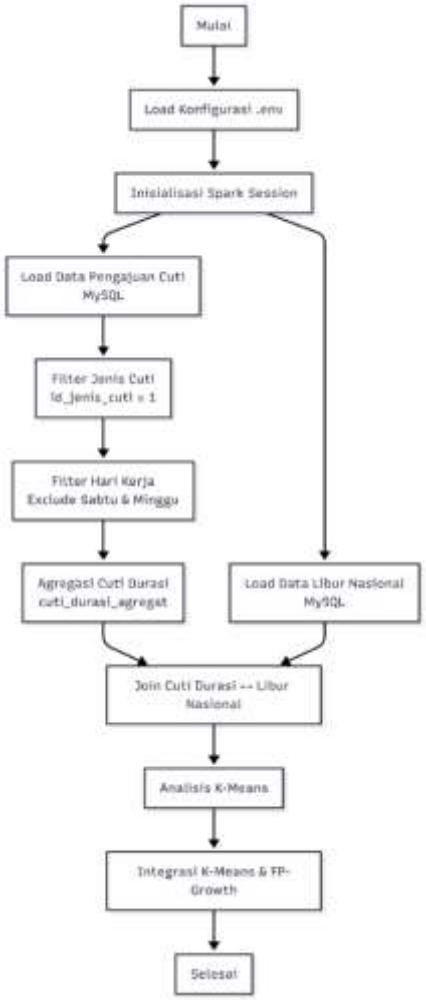

Figure 1. Research Flowchart

The figure illustrates the research workflow in analyzing the correlation between leave applications and national holidays within the context of personnel governance. Utilizing structured personnel data is considered paramount to providing empirical insights into employee leave-taking behaviors and their implications for service planning and human resource management within the workplace.

This study contributes by processing large-scale leave data through the Extraction, Transformation, and Loading (ETL) stages using Apache Spark. During the transformation stage, the initial leave application data, originally formatted as date ranges is not permanently stored as daily leave records. Instead, it is temporarily expanded to a daily level for the purpose of calculating duration attributes. This temporary expansion is utilized to calculate the total number of calendar days, effective working days, weekends, and national holidays falling between the start and end dates of the leave, thereby yielding a more accurate representation of leave duration that aligns with the institution's actual operational conditions.

Furthermore, the transformed leave data is linked with national holiday data sourced from the official holiday table, which includes identifying relative temporal positions such as before the holiday (H-1), during the holiday (H), and after the holiday (H+1). This process aims to identify the temporal proximity between leave periods and national holidays without evaluating individual employees directly. The processed data is then re-stored into the database as an analytical table and utilized for advanced analysis, including temporal aggregation analysis, clustering of leave application patterns, and the visualization of leave-taking tendencies relative to national holiday periods. This big data analytics-based temporal analysis approach aligns with previous research highlighting that large-scale data processing is capable of identifying latent data trends and tendencies (Chen & Storey, 2012).

Data and Data Sources

The data utilized in this study constitutes primary administrative data obtained from the employee leave application information system across several government agencies in Semarang City. The data originates from the application table within the E-Cuti Application, which is managed centrally and stored in a MySQL database. The dataset spans from 2020 to 2025, comprising approximately 150,000 records that contain information on employee identities, working units, leave types, and the start and end dates of the requested leave.

This dataset was selected due to its extensive longitudinal timeframe, large volume, and adequate level of completeness and consistency, thereby making it highly representative for accurately capturing employee leave application patterns. Moreover, utilizing data sourced directly from the official operational systems of these agencies is expected to enhance the reliability of the analysis. As supporting data, this study also utilizes national holiday and collective leave (*cuti bersama*) data obtained from official municipal sources in Semarang to facilitate the temporal correlation analysis between leave applications and holiday periods. All data utilized has undergone an anonymization process and is strictly used for research purposes, ensuring that no personal employee identities are directly disclosed.

Table 1. Leave Application Data Attributes

Code	Attribute	Description
A1	id_cuti	Leave Application ID (Unique Identifier)
A2	nama_pegawai	Employee Name
A3	nip_pegawai	Employee National Identification Number (*Nomor Induk Pegawai*)
A4	tanggal_mulai_diajukan	Start date of the requested leave
A5	tanggal_selesai_diajukan	End date of the requested leave
A6	id_jenis_cuti	Leave Type ID
A7	opd	Regional Government Organization

		(*Organisasi Perangkat Daerah*)
A8	unit_kerja	Employee's designated working unit
A9	lamanya_cuti	Leave duration
A10	status	Approval status (e.g., Approved, Rejected, Pending)

Architecture and Data Processing Environment

This study was executed within a computing environment powered by the Windows 11 64-bit operating system, equipped with specialized software for large-scale data processing. Python version 3.10 was utilized as the primary programming language for developing the analytical scripts, while Apache Spark version 3.5.7 via the PySpark interface was leveraged to perform distributed data processing. The MySQL Server version 8 Relational Database Management System (RDBMS) served as the repository for both the raw source data and the final analytical results, interconnected via the MySQL Connector/J version 8.0.33.

Additionally, the Pandas library was employed for tabular data manipulation during advanced visualization phases, while Matplotlib was utilized to generate the analytical graphical outputs. This integrated software stack was selected due to its robust capabilities in handling large-scale data computing efficiently, enabling seamless integration across the ETL pipeline, analytical processing, and data visualization phases.

Table 2. Research Software Configuration

No	Software / Library / Driver	Version	Role / Function in Research
1	Windows Operating System	11 (64-bit)	Main host computing environment
2	Python	3.10	Core programming language for script development
3	Apache Spark (PySpark)	3.5.7	Distributed parallel data computing and ETL processing framework
4	MySQL Server	8.0	Source and target database management system (RDBMS)
5	MySQL Connector/J	8.0.33	JDBC driver bridging Apache Spark and MySQL database connectivity
6	Pandas	Latest / Stable	Tabular data manipulation for advanced post-processing
7	Matplotlib	Latest / Stable	Graphical data visualization tool for analytical results

The ETL (Extract, Transform, Load) Process

The initial phase of analysis in this study is executed through the Extract, Transform, and Load (ETL) pipeline utilizing the DataFrame-based Apache Spark framework. These three stages are paramount in managing large-scale datasets, particularly within complex big data environments (Informatika & Lunak, 2025). The **Extract** stage is carried out by ingesting the employee leave application data directly from the MySQL relational database via a JDBC connection. The extracted data is filtered specifically for annual leave applications while ensuring the completeness and consistency of both the start and end dates of the leave.

In addition to the leave application data, national holiday data is also retrieved during the extraction stage from the libur_resmi table stored within the same database. This table contains official holiday dates along with their respective descriptions, serving as a temporal reference to identify national holidays occurring within, prior to, or following the requested leave period. Utilizing official holiday data from an independent source aims to ensure that leave duration calculations are performed objectively and in accordance with the applicable national calendar.

During the **Transform** stage, the initial leave application data, originally formatted as date ranges is processed to compute detailed leave duration attributes. This calculation leverages temporary date expansion via the sequence and explode functions in Apache Spark. Rather than generating permanent daily data storage, this expansion is utilized dynamically to calculate the number of effective working days, weekends, and national holidays falling between the start and end dates of the leave by executing a join operation against the libur_resmi table. Saturdays, Sundays, and national holidays are explicitly isolated so that the computed leave duration accurately reflects the actual operational conditions of the institution.

The **Load** stage is executed by saving the transformation outputs into an analytical table within the MySQL database, specifically designated as the cuti_durasi_analitik table. Each row within this target table represents a single leave application that has been enriched with metrics regarding the number of working days, weekends, national holidays, descriptions of official holidays intersecting the leave period, and the total calendar duration. Through this rigorous ETL pipeline, the initial leave application dataset of approximately 150,000 records was filtered and refined down to 82,846 valid and highly representative records for advanced analytical processing.

Table 3. Structure of the cuti_durasi_analitik Table

Code	Attribute	Description
A1	id_pengajuan	Leave Application ID (INT)
A2	nip	Employee National Identification Number / *NIP* (BIGINT)
A3	opd	Regional Government Organization / *OPD* (LONGTEXT)
A4	unit_kerja	Employee's designated working unit (LONGTEXT)
A5	lokasi_kerja	Physical location of the workplace (LONGTEXT)
A6	tanggal_mulai_diajukan	Start date of the requested leave (DATE)
A7	tanggal_selesai_diajukan	End date of the requested leave (DATE)
A8	jumlah_hari_kerja	Leave duration during official working days (INT)
A9	jumlah_hari_weekend	Number of weekend days within the leave period (INT)
A10	jumlah_hari_libur_resmi	Number of official public holidays within the leave period (INT)
A11	libur_resmi_diantara	Official holidays intersecting the leave period (LONGTEXT)
A12	total_hari_kalender	Total leave duration calculated in calendar days (INT)

III.RESULTS AND DISCUSSIONS

Temporal Aggregation Analysis

Temporal aggregation analysis is conducted to obtain a comprehensive overview of employee leave application patterns based on specific time dimensions and units of analysis. This process initiates by aggregating the total number of employees taking leave on each working date, thereby generating daily leave application density metrics as the foundation for temporal analysis. This time-based aggregation is utilized to summarize massive individual records into a time-series format that represents the collective dynamics of a phenomenon over a specific period. This approach facilitates the identification of trends, fluctuations, and seasonal patterns emerging from historical data, making it highly relevant for analyzing organizational behaviors and activities that possess a strong time dimension, such as administrative and personnel data (Chen & Storey, 2012).

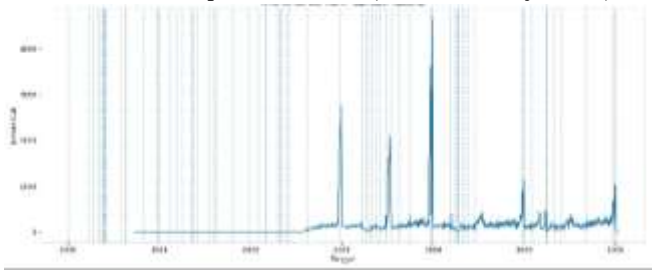

Figure 2. Visualization of Aggregated Leave Applications by Date and Correlation with Public Holidays

In this study, temporal aggregation is not only executed at the daily level but is further extended to multiple aggregation levels as required by the analytical objectives. The transformed data is utilized to construct the cuti_agregat_nip_tahunan table, which represents an annual summary of leave applications for each employee (NIP). This aggregation encompasses metrics such as the total number of leave applications, the cumulative working days of leave taken within a year, and the average annual leave duration per employee. This time-based aggregation approach enables the analysis of long-term individual behavioral tendencies and facilitates the identification of shifting patterns over time within administrative datasets (Chen & Storey, 2012).

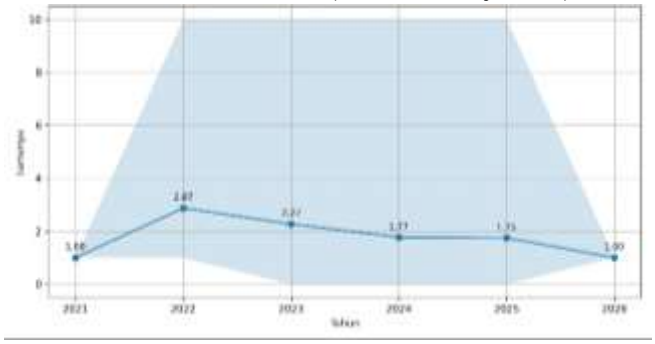

Figure 3. Visualization of Average Leave Application Duration Per Year

Subsequently, aggregation based on organizational units is executed to capture variations in leave patterns at a structural level. The cuti_agregat_opd and cuti_agregat_unit_kerja tables are constructed by grouping the leave data according to Regional Government Organizations (*OPD*) and specific working units. This structural aggregation aims to identify differences in the intensity and characteristics of leave applications across organizational units, which may reflect variations in workload dynamics, public service functions, or the respective operational characteristics of each unit.

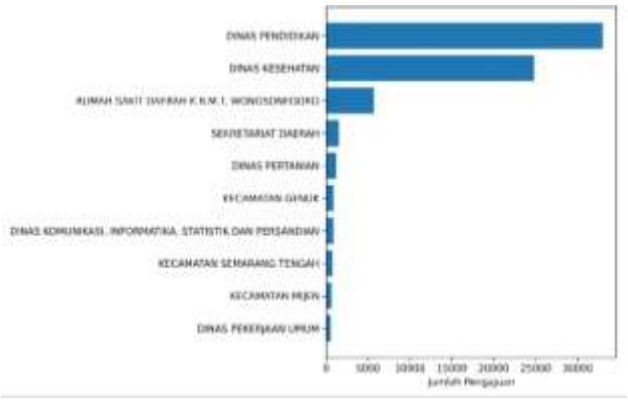

Figure 4. Visualization of the Top Ten Regional Organizations with the Highest Number of Leave Applications

At the macro level, aggregation is also performed on an annual basis through the cuti_agregat_tahunan table, which presents a summary of the total number of leave applications and the average leave duration per year. This annual aggregation is utilized to observe long-term trends in employee leave applications as well as shifting leave patterns across different years, particularly in relation to national holiday periods and collective leave (*cuti bersama*). By leveraging multiple levels of temporal and organizational aggregation, the analysis is capable of providing a comprehensive overview of employee leave-taking patterns from individual, working unit, and macro-organizational perspectives (Nurhayati & Prabowo, 2022).

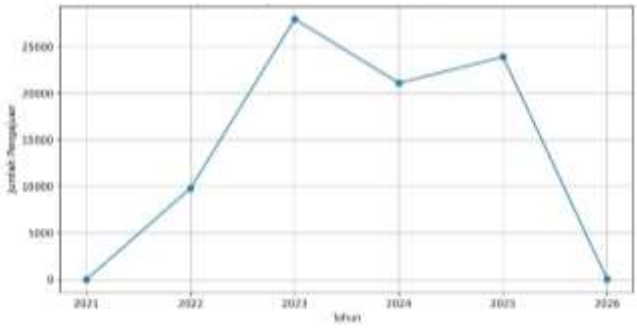

Figure 5. Visualization of the Number of Leave Applications Per Year

Clustering of Leave Application Patterns

Clustering of leave application patterns is performed to categorize employee leave requests based on their temporal characteristics and timing contexts, particularly their intersection with working days, weekends, and national holidays. The data utilized at this stage is derived from the prior preprocessing phase stored in the cuti_durasi_analitik table. Within this table, each data row represents a single leave application enriched with metrics such as calendar duration, number of effective working days, number of weekends, and the frequency and descriptions of official public holidays intersecting the leave period. The clustering approach is employed to group leave application patterns based on the aggregate similarity of data characteristics, without evaluating or profiling individual employees. Consequently, the analysis remains strictly objective, focuses on collective group patterns, and avoids discriminatory interpretations at the individual level, in accordance with the fundamental principles of cluster-based data analysis (Chen & Storey, 2012).

The clustering process in this study utilizes the **K-Means** algorithm, implemented via the Apache Spark MLlib library. The procedural sequence of the K-Means clustering algorithm to successfully partition data into multiple clusters begins by defining the K value as the desired number of clusters and establishing initial centroids for each respective cluster (Suryadi et al., 2020). This algorithm was selected due to its high computational efficiency and its robust capability to process large-scale datasets effectively, making it highly suitable for analyzing employee leave application data with a significant volume of records.

The variables selected as the foundation for clustering represent behavioral characteristics of leave-taking. These features include whether the leave is taken exclusively on official working days, whether the leave period falls immediately before or after a national holiday, whether the leave straddles or sandwiches a holiday, whether the leave spans from Monday to Friday, the presence of weekends within the leave timeframe, and the total number of effective working days requested. All these variables are numerically encoded and consolidated into a unified feature vector using Spark's Vector Assembler. Subsequently, feature standardization is executed prior to the clustering process to prevent variables with larger scales from dominating the distance metrics.

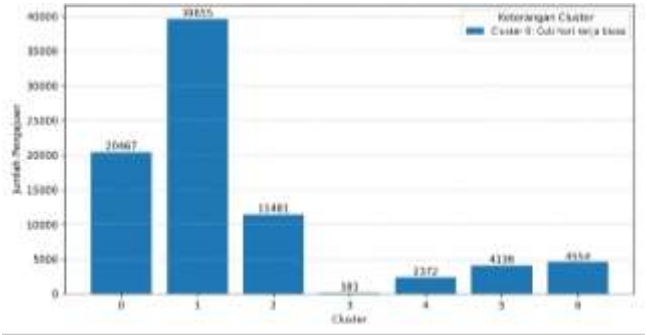

Figure 6. Number of Leave Applications Per Cluster

The number of clusters in this study is established at seven (K = 7) to capture a more diverse and contextual variation of employee leave-taking behavioral patterns. This optimal cluster count was determined by evaluating the underlying complexity of leave patterns in relation to official working days, weekends, and national holidays. The clustering results demonstrate distinct and well-defined characteristics across each segment, reflecting varied tendencies among employees in utilizing their leave. These include patterns such as leave taken exclusively during regular working days, leave that straddles or sandwiches national holidays, and long-duration leave that intersects weekends. A summary of the distinct characteristics of each cluster is presented in Table 4, serving as the baseline for subsequent advanced analysis and the

institutional interpretation of employee leave management policies.

Table 4. Clustering of Leave Types

Cluster	Pattern Description
0	Regular working day leave (*Normal baseline leave*)
1	Post-holiday leave (*Leave taken immediately after a public holiday*)
2	Pre-holiday leave (*Leave taken immediately before a public holiday*)
3	Holiday-straddling leave (*Sandwich day / "Harapitnas" leave pattern*)
4	Monday-initiated leave (*Leave starting on the first day of the workweek*)
5	Friday-terminated leave (*Leave ending on the last day of the workweek*)
6	Weekend-straddling leave (*Extended leave bridging across Saturdays and Sundays*)

Analisis Frequent Pattern

As an advanced analysis following the clustering phase, this study implements the Frequent Pattern Growth (FP-Growth) algorithm. FP-Growth is a prominent alternative algorithm utilized to determine the most frequently occurring datasets (*frequent itemsets*) within a large data repository (Wilda et al., 2025). While previous studies have leveraged FP-Growth for optimizing product placement layout (Munanda & Monalisa, 2021), identifying travel ticket sales patterns (Wilda et al., 2025), and discovering coffee sales associations at retail outlets (Juwita, 2024), this study uniquely applies the algorithm to identify recurrent combinations of behavioral leave-taking characteristics. The data utilized at this stage is sourced from the cuti_cluster_kmeans table, where each row represents an individual leave application that has been classified into a specific cluster and enriched with a series of behavioral leave attributes. In this context, each leave application is treated as a single transaction comprising a set of items that reflect the temporal characteristics of the leave, such as leave taken exclusively during working days, leave positioned immediately before or after a national holiday, holiday-straddling leave, and the inclusion of weekends within the leave period.

The FP-Growth algorithm is employed to extract the most frequently occurring attribute combinations without explicitly generating candidate itemsets. This architectural approach renders FP-Growth significantly more computationally efficient than the traditional Apriori algorithm, particularly when scaled to massive datasets with high transaction volumes (Larasati et al., 2015). Within the framework of this research, FP-Growth facilitates the identification of aggregate and repetitive leave behavior patterns without evaluating or profiling individual employees.

The transaction generation process is conducted by converting behavioral leave attributes into discrete itemsets, where each active attribute value is represented as an individual item. Subsequently, the FP-Growth algorithm is executed using predefined minimum support and minimum confidence thresholds to generate frequent itemsets that represent the most dominant leave behavior combinations. The analytical results reveal specific associative patterns, such as the tendency for leave to be taken exclusively on working days proximate to national holidays, or patterns where leave straddles official holidays while simultaneously intersecting a weekend.

The findings derived from this frequent pattern analysis offer an extended perspective in understanding the collective temporal dynamics of employee leave applications. These identified patterns serve as an empirical baseline for evaluating leave management frameworks and workload planning strategies, specifically in forecasting and mitigating periods with a high potential density of leave applications surrounding national holidays.

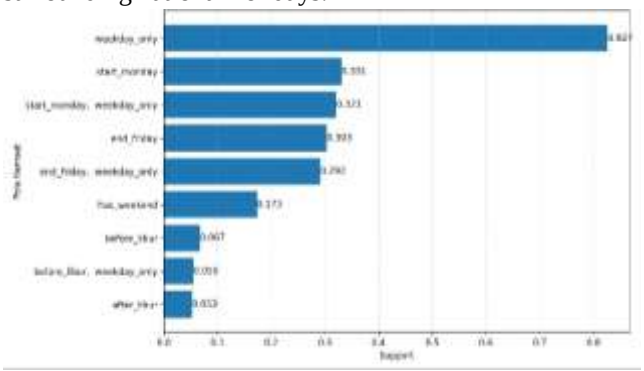

Figure 7. Top 10 Leave Application Patterns Based on Correlation with Public Holidays

Integration of K-Means and FP-Growth

The integration of K-Means clustering results and FP-Growth frequent pattern analysis is conducted to obtain a more comprehensive understanding of employee leave-taking behaviors. K-Means is utilized to categorize leave applications based on the similarity of their temporal characteristics and calendar contexts, while FP-Growth serves to identify recurrent combinations of leave behavior attributes across the entire dataset.

The integration results demonstrate that clusters exhibiting a tendency for leave-taking proximate to national holidays have a strong association with frequent itemsets involving the before_libur, after_libur, and mengapit_libur attributes. Conversely, the regular working-day leave cluster is heavily dominated by the weekday_only attribute and shows no significant correlation with national holidays.

By interconnecting the clustering outputs with frequent itemsets, each cluster can be interpreted substantively rather than merely existing as a numerical group. The integration of K-Means and FP-Growth enhances the validity of the exploratory analysis because the patterns emerging within the clusters are empirically supported by frequently occurring

item combinations. This dual approach is highly relevant for analyzing large-scale administrative datasets and can be utilized as an empirical baseline for evaluating leave management frameworks and workload planning strategies, without shifting toward individual tracking or automated decision support systems.

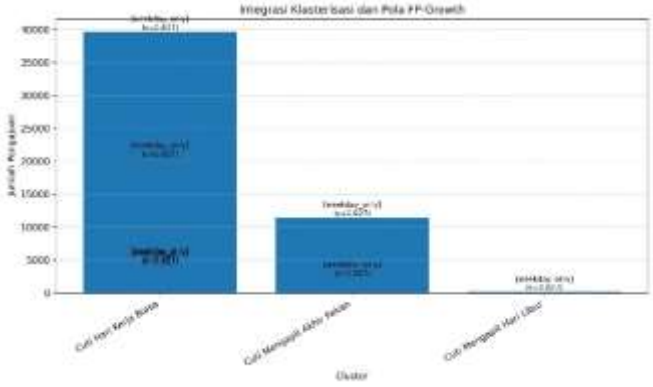

Figure 8. Integration of Clustering and FP-Growth Patterns

IV. CONCLUSIONS

This study implements Big Data Analytics powered by the Apache Spark framework to analyze employee leave application patterns in Semarang City spanning the 2020–2025 period. Out of approximately 150,000 raw leave records, the ETL pipeline yielded 82,846 valid applications after rigorous filtering based on annual leave types, complete date attributes, and logical duration consistency. The transformed dataset incorporates critical metrics such as calendar duration, effective working days, weekend counts, and the number of national holidays intersecting the leave period, thereby enabling a structured and highly efficient temporal analysis.

The clustering results generated by the K-Means algorithm successfully partitioned the leave application behaviors into seven distinct clusters, with the average leave duration ranging between 1.74 and 2.35 days. The cluster with the highest application frequency represents leave taken exclusively during regular working days. Conversely, clusters involving weekends or national holidays exhibit relatively longer calendar durations but are characterized by lower application volumes.

Furthermore, frequent pattern mining analysis using the FP-Growth algorithm reveals that the pure working-day leave pattern (weekday_only) holds the highest support value at 0.827. Other associative patterns related to national holidays and weekends, such as leave initiated or terminated adjacent to a holiday or leave straddling a weekend were successfully identified but yielded significantly lower support values. These findings indicate that proximity to national holidays and weekends exerts a limited influence on aggregate leave application behaviors. Consequently, employee leave-taking tendencies remain predominantly normative and are not heavily dominated by the strategic exploitation of holiday periods.

REFERENCES

1. Michael Armbrust, Reynold S. Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K. Bradley, Xiangrui Meng, Tomer Kaftan, Michael J. Franklin, Ali Ghodsi, and Matei Zaharia. 2015. Spark SQL: Relational Data Processing in Spark. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD '15). Association for Computing Machinery, New York, NY, USA, 1383–1394. https://doi.org/10.1145/2723372.2742797
2. Chen, Hsinchun & Chiang, Roger & Storey, Veda. (2012). Business Intelligence and Analytics: From Big Data to Big Impact. MIS Quarterly. 36. 1165-1188. 10.2307/41703503.
3. Han, Jiawei & Kamber, Micheline & Pei, Jian. (2012). Data Mining: Concepts and Techniques. 10.1016/C2009-0-61819-5.
4. OECD. (2020). The OECD Digital Government Policy Framework: Six dimensions of a Digital Government. Paris : OECD Publishing. https://doi.org/10.1787/f64fed2a-en
5. Hadad, S. H. (2024). Analisis Prioritas Pemberian Cuti Karyawan Menggunakan Metode Pembobotan Entropy dan Simple Multi Attribute Rating Technique. Journal of Artificial Intelligence and Technology Information (JAITI), 2(2), 106-117. https://doi.org/10.58602/jaiti.v2i2.126
6. Laksono, et al.. (2025). Perbandingan Apache Airflow dan Apache Spark dalam Proses ETL untuk Memprediksi DropOut dan Keberhasilan Akademik Mahasiswa. Jurnal Informatika dan Rekayasa Perangkat Lunak, 7(2).
7. Larasati, D. P., Nasrun, M., Si, S., & Ahmad, U. A. (2015). *ANALISIS DAN IMPLEMENTASI ALGORITMA FP-GROWTH PADA APLIKASI SMART UNTUK MENENTUKAN MARKET BASKET ANALYSIS PADA USAHA RETAIL (STUDI KASUS : PT . X) ANALYSIS AND IMPLEMENTATION OF FP-GROWTH ALGORITHM IN SMART APPLICATION TO DETERMINE MARKET BASKET ANALYSIS ON RETAIL BUSINESS (CASE STUDY : PT . X)*. *2*(1), 749–755.
8. Ines Mergel, Noella Edelmann, Nathalie Haug, Defining digital transformation: Results from expert interviews, Government Information Quarterly, Volume 36, Issue 4, 2019, 101385, ISSN 0740-624X, https://doi.org/10.1016/j.giq.2019.06.002.
9. Mudjiyanto, Bambang. (2018). TIPE PENELITIAN EKSPLORATIF KOMUNIKASI. Jurnal Studi Komunikasi dan Media. 22. 65. 10.31445/jskm.2018.220105.
10. 1Elvira Munanda, 2Siti Monalisa, PENERAPAN ALGORITMA FP-GROWTH PADA DATA

11. TRANSAKSI PENJUALAN UNTUK PENENTUAN TATALETAK, Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi, Vol. 7, No. 2, Agustus 2021, Hal. 173-184, e-ISSN 2502-8995 p-ISSN 2460-8181
12. Azzahra Nur Oktavia, Iqbal Muhammad, Rikky Wahyu Saputra, Muhammad Ichsan Zulfikar, Aries Saifudin. Implementasi Metode Natural Language Processing Dalam Studi Analisis Semantik Dan Emosi Buzzer Pada Tweet Di Aplikasi X, Buletin Ilmiah Ilmu Komputer dan Multimedia (BIIKMA), 2024, pp. 154-159.
13. Sapta Hastho Ponco*, Susanti, Maziyyatul Mufiedah, A Big Data Approach to Mapping Local Craft Trends: A Study of Creative Economy Opportunities through Online Search Activity in Indonesia, Seminar Nasional Official Statistics 2025.
14. Juwita, Ita & Ali, Irfan. (2024). PENERAPAN POLA PENJUALAN DENGAN MENGGUNAKAN METODE ALGORITMA ASOSIASI FP-GROWTH BERTUJUAN UNTUK MENINGKATKAN PENJUALAN KOPI DI POINT COFFEE. JATI (Jurnal Mahasiswa Teknik Informatika). 8. 1600-1607. 10.36040/jati.v8i2.9025.
15. Suryadi, Usep T., and Sri Saraswati. "Sistem Cerdas Pemantau Kenyamanan Ruang Kelas Berbasis Internet of Things (Iot) Menggunakan Metode K-means Pada Platform Thingspeak." Jurnal Teknologi Informasi dan Komunikasi, vol. 15, no. 1, 1 Apr. 2020, pp. 70-81.
16. Utami, F.D., & Astuti, F.D. (2024). Comparison of Hadoop Mapreduce and Apache Spark in Big Data Processing with Hgrid247-DE. Journal of Applied Informatics and Computing.
17. Wilda, Roudotul & Saripurna, Darjat & Sulaiman, Oris. (2025). Implementasi Algoritma Frequent Pattern Growth (FP-Growth) untuk Pola Penjualalan Tiket Travel pada PT Taxi Kita Bersama. Hello World Jurnal Ilmu Komputer. 3. 138-145. 10.56211/helloworld.v3i3.588.
18. Zaharia, Matei & Xin, Reynold & Wendell, Patrick & Das, Tathagata & Armbrust, Michael & Dave, Ankur & Meng, Xiangrui & Rosen, Josh & Venkataraman, Shivaram & Franklin, Michael & Ghodsi, Ali & Gonzalez, Joseph & Shenker, Scott & Stoica, Ion. (2016). Apache spark: A unified engine for big data processing. Communications of the ACM. 59. 56-65. 10.1145/2934664.