Jurnal Teknik Informatika C.I.T Medicom, 17 (2) (2025) 59-74

Published by: Institute of Computer Science (IOCS)

Jurnal Teknik Informatika C.I.T Medicom

Journal homepage: www.medikom.iocspublisher.org

Smart City Weather and Disaster Monitoring Architecture: LoRaWAN Integration with COBIT 2019 Governance

Bambang Saras Yulistiawan [1], Galih Prakoso Rizky A [2], Rifka Widyastuti [3], RR Octanty Mulianingtyas [4]

[1), 2), 3), 4)] Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jakarta, Indonesia

Article Info

Article history

Received : Jun 20, 2025
Revised : Jun 30, 2025
Accepted : Jul 30, 2025

Keywords:

LoRaWAN;
COBIT 2019;
Smart City;
Pemantauan Bencana,
IoT.

Abstract

Climate change, urbanization, and the increasing frequency of natural disasters such as floods and forest fires demand that Indonesian cities adopt real-time, integrated, and reliable environmental monitoring systems. Within the context of smart cities, LoRaWAN technology offers wide coverage, low power consumption, and cost-efficient operations, making it highly relevant for city-scale multi-sensor monitoring systems. This study proposes the design of a LoRaWAN-based weather and disaster monitoring system architecture integrated into the smart city framework, while simultaneously adopting the IT governance principles of COBIT 2019. The methodology includes a literature review and the mapping of five COBIT domains (EDM03, APO03, BAI03, DSS02, MEA01) to LoRaWAN's technical components, ranging from sensors, gateways, and network servers to application servers, dashboards, and public notification modules. The analysis demonstrates that the proposed design enhances data standardization, end-to-end security, monitoring, scalability, and device governance. The integration of COBIT 2019 further enables the optimization of risk management, monitoring effectiveness, incident response, and regulatory compliance. In conclusion, the proposed architecture provides a comprehensive framework to support resilient, adaptive, and sustainable smart cities. However, this architecture has not yet been implemented in practice, thus necessitating further implementation and evaluation to ensure the system's effectiveness and sustainability in operational environment.

Corresponding Author:

Bambang Saras Yulistiawan,
Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
Jl. Rumah Sakit Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, Indonesia, 12450
Email: bambangsarasyulistiawan@upnvj.ac.id

1. Introduction

Climate change and rapid urbanization have increased the vulnerability of cities worldwide, including those in Indonesia, to various natural hazards such as floods, forest fires, and extreme weather events. Major cities such as Jakarta, Surabaya, and Palembang routinely face the risk of annual flooding, which disrupts community activities and causes significant economic losses. Meanwhile, forest fires in Kalimantan and Sumatra have further exacerbated environmental degradation and posed serious threats to public health [7].

In response to these challenges, the smart city concept has emerged as a strategic solution that integrates information and communication technology (ICT) to enhance the efficiency of public services and improve the quality of life [28], [4], [36]. A critical component in smart city development is the capability to conduct real-time environmental monitoring, thereby enabling timely and effective decision-making in disaster risk mitigation.

The Internet of Things (IoT) offers substantial opportunities to build sensor-based environmental monitoring systems at scale. Among the various IoT communication technologies, LoRaWAN (Long Range Wide Area Network) stands out for its advantages in wide communication coverage, low power consumption, and cost-efficient operation [1], [9]. LoRaWAN enables the integrated collection of environmental data from diverse sensors, including temperature, humidity, rainfall, wind speed, and flood sensors. A study by Aji and Nugroho [26] demonstrated that the implementation of LoRaWAN can significantly improve the effectiveness of early warning systems for urban disasters.

However, the effectiveness and sustainability of LoRaWAN-based weather and disaster monitoring systems are determined not only by technical aspects but also by sound information technology (IT) governance. COBIT 2019, as an internationally recognized IT governance framework, provides guidance to ensure alignment between technological solutions, organizational needs, regulatory compliance, and data security [2]. Research by S. Wang, L. Zhang, et al. [27] highlights that structured IT governance is essential to the success of smart city initiatives, particularly in ensuring security, effectiveness, and the sustainability of public services.

Building on this background and related studies, this research develops an architecture for a weather and disaster monitoring system that integrates LoRaWAN into smart city infrastructure, while adopting IT governance principles based on COBIT 2019. This development is expected to contribute to the establishment of an effective, secure, sustainable, and adaptive monitoring system for addressing disaster challenges in the digital era.

2. Research Methodology

This study employs a descriptive qualitative approach combined with system engineering to develop and analyze a LoRaWAN-based weather and disaster monitoring architecture integrated with COBIT 2019 standard IT governance within the smart city framework. The research methodology consists of several key stages as follows.

2.1 Literature Review and Requirements Analysis

The study conducted by Aji and Nugroho [5] highlights the effectiveness of LoRaWAN-based disaster monitoring systems in urban environments and provides empirical evidence regarding sensor integration and data analysis processes relevant to local needs. Meanwhile, the COBIT 2019 framework developed by ISACA [2] is employed to ensure that the proposed system adheres to IT governance standards, data security, and sustainability in managing smart city data and business processes.

Furthermore, early-generation LoRaWAN design studies documented by L. Chen, M. Rossi, et al. [31], Bor et al. [14], P. S. Kumar, T. H. Lee [32], and Adelantado et al. [12] discuss the limitations of previous monitoring system architectures and data pipelines, such as radio coverage, uplink capacity, and the absence of governance features. Analysis of these limitations serves as the basis for identifying research gaps, which in turn form the foundation for architectural innovation in this study.

Insights into smart city requirements in Indonesia are drawn from the works of Heeks [3], Indrajit [4], and Suryono et al. [8], which emphasize the importance of IT governance in supporting the effectiveness, security, and sustainability of public services. In the context of national regulation, the Head of BNPB Regulation No. 4 of 2008 [7] is used as a reference to ensure that the proposed monitoring system complies with the guidelines and procedures for disaster management in Indonesia.

Accordingly, the combination of literature used at this stage not only strengthens the theoretical and technical foundation for architectural development but also ensures relevance and compliance with both national and international requirements and regulations.

2.2 System Architecture Design

The system architecture design is developed based on the latest LoRaWAN specifications [1] and the principles of information technology governance defined in the COBIT 2019 framework [2]. In the design process, the COBIT domains EDM (Evaluate, Direct, and Monitor), APO (Align, Plan, and Organize), BAI (Build, Acquire, and Implement), DSS (Deliver, Service, and Support), and MEA (Monitor, Evaluate, and Assess) are mapped to the technical components of LoRaWAN. This mapping represents a crucial step to ensure the integration of IT governance with technological operations. The EDM domain emphasizes the evaluation, direction, and monitoring of IT policies and performance; APO focuses on aligning strategies, planning, and organizing resources; BAI encompasses the development and implementation of technological solutions; DSS concentrates on service delivery, operational support, and data security; while MEA ensures continuous monitoring, evaluation, and assessment of both system performance and compliance. Through this mapping, each architectural element ranging from edge sensors, gateways, network servers, and join servers to application servers, integration layers, and public notification modules can be systematically integrated within a sustainable governance framework in accordance with established standards. The architectural visualization is represented in the form of a reference diagram that illustrates data flow, transport security mechanisms, and application integration to support the effectiveness and security of disaster monitoring systems in smart city environments.

2.3 Comparative Analysis and Validation

The comparative analysis in this study was conducted by critically reviewing previous research on the design and implementation of early-generation LoRaWAN. Augustin et al. [11] discussed the key characteristics of LoRa as a long-range, low-power communication technology, as well as challenges related to network scalability and interference management in high-density node environments. Bor et al. [14] highlighted the limitations of uplink capacity and potential gateway bottlenecks, which affect system performance when the number of devices increases significantly.

Studies by Ferrari et al. [29]and Zhang et al. [30] provided an in-depth exploration of LoRaWAN's limitations, particularly in terms of channel access, data transmission mechanisms, and end-to-end security. Their research also identified the need for standardized data models and device governance to enable efficient integration with smart city infrastructure. Centenaro et al. [10] and Mekki et al. [13] expanded the analysis by comparing the performance of LoRaWAN with other LPWAN technologies, such as Sigfox and NB-IoT, while also discussing interference mitigation strategies for unlicensed frequency bands to ensure reliable communication in dense device environments.

Sanchez-Iborra and Cano [15] examined security aspects as well as monitoring and observability mechanisms in LoRaWAN systems, including the challenges of implementing multi-channel operations to enhance throughput without compromising quality of service. Recent studies by Wang et al. [33] and Kumar et al. [34] examined security aspects as well as monitoring and observability mechanisms in LoRaWAN systems, including the challenges of implementing multi-channel operations to enhance throughput without compromising quality of service. Mikhaylov et al. [16] reviewed scalability and reliability issues in large-scale LoRaWAN deployments, emphasizing the need for adaptive device governance to ensure optimal system management in line with industry standards.

Based on these studies, the comparative analysis in this research focuses on data model standardization, end-to-end security, observability, scalability, device governance, and multi-channel integration. The proposed architecture was validated by mapping system components to COBIT governance domains and to the specific requirements of disaster monitoring in smart city environments.

This step ensures that the developed architecture addresses the challenges identified in earlier studies while providing a more efficient, secure, and integrated system in accordance with applicable IT governance standards.

2.4 Synthesis and Implication

The synthesis of the architectural analysis and validation indicates that the LoRaWAN-based weather and disaster monitoring system, when integrated with COBIT 2019 IT governance principles, demonstrates several key advantages. First, this integration ensures standardization in data models, secure communication flows, and structured management of devices and networks in alignment with IT governance domains. Second, the mapping of COBIT domains EDM, APO, BAI, DSS, and MEA onto LoRaWAN technical components provides a systematic framework for risk management, strategic planning, service development, and continuous system performance evaluation.

The main contribution of this research lies in the proposal of an architectural model that is adaptive to the needs of Indonesian smart cities, while offering a technological solution that enhances disaster monitoring effectiveness, strengthens data security, and facilitates multi-channel integration for system scalability. The proposed model also serves as a reference for implementing IT governance in city-level IoT systems, thereby offering practical guidance for local governments and smart city developers.

Nevertheless, this study identifies several challenges, including the need for greater network capacity as device numbers and data volume increase, potential interference in unlicensed frequency bands, and the necessity of harmonizing regulations and security standards between government authorities and technology providers.

The implications of this research highlight new opportunities for developing effective, secure, and sustainable smart cities through the application of LoRaWAN integrated with IT governance. Local governments are encouraged to adopt this architecture to strengthen disaster monitoring systems and improve emergency responsiveness, while ensuring sustainability and compliance with national regulations on disaster management and information technology governance. Finally, the research methodology is expected to produce not only an architectural model but also an IT governance analysis that can serve as a reference for future development of disaster monitoring systems in smart city environments.

3. Result and Discussion

In this study, a series of stages in the design and development of a LoRaWAN-based weather and disaster monitoring system architecture integrated with the COBIT 2019 IT governance standard were carried out. This integration aims to produce a system architecture that not only fulfills the aspects of sustainability and effectiveness but also ensures compliance with governance principles, security, and scalability required for implementation within smart city environments.

3.1 Sensor Architecture Design and End-Node

At the system design stage, the end-node component functions as the primary point for collecting environmental and disaster-related data, which are then transmitted through the LoRaWAN network. The sensors integrated into the end node include temperature and humidity sensors (DHT22 or SHT31), air pressure sensors (BMP280 or BME280), rainfall sensors, hazardous gas sensors (MQ-135), vibration or seismic sensors (accelerometers such as ADXL345), and water level sensors for flood detection. The selection of these sensors is based on the requirements of multi-hazard monitoring in urban environments and refers to standard disaster monitoring guidelines [5], [7].

Each end node is designed using low-power microcontrollers such as Arduino, ESP32, or STM32, integrated with a LoRa module (SX1276/78) and a power management system supported by batteries

and solar panels, enabling long-term operation without reliance on the main power grid. The data collected by the sensors are processed and packaged into a standardized payload format before being transmitted to the LoRaWAN gateway.

The end-node design also considers device governance and data security aspects. Each device is assigned a unique identity and employs authentication protocols as well as data encryption in accordance with the DSS and MEA domains of the COBIT 2019 framework [2]. Regular device health monitoring is performed to ensure the availability and reliability of the monitoring system.

Based on the design and mapping to established standards, the end nodes are capable of detecting and transmitting environmental data in real time with efficient power consumption. The integration of multiple sensor types within the end node enables the system to simultaneously monitor various parameters, thereby conceptually enhancing the effectiveness of early detection and disaster response in smart city environments. The architecture design is illustrated in Figure 1. The validity of the system's performance and effectiveness is grounded in simulation, design analysis, and related literature, while further testing is required to confirm actual performance in the field.

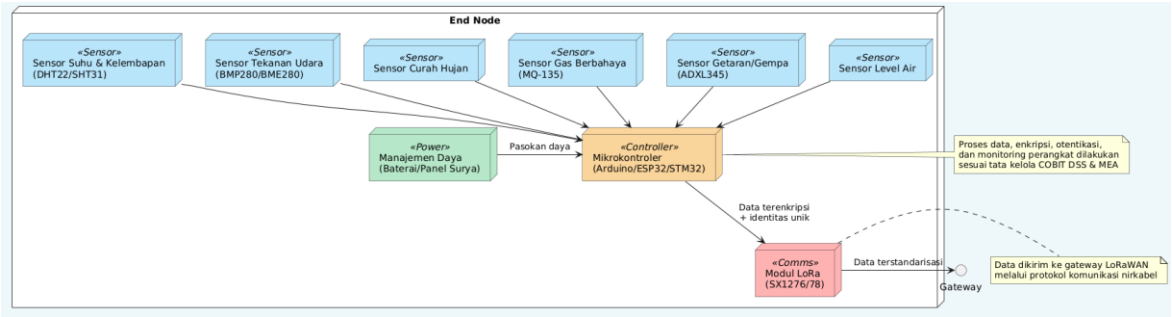

Figure 1. End Node Architecture.

3.2 Architecture Design of Gateway and Network Server

To provide a clearer picture of the communication flow and security mechanisms in the LoRaWAN system, the following data flow architecture is shown from the Sensor End Node to the Network Server. This diagram highlights the function of each component, including gateways and network servers, as well as the implementation of security protocols, logging processes, performance monitoring, and compliance with regulations such as COBIT.

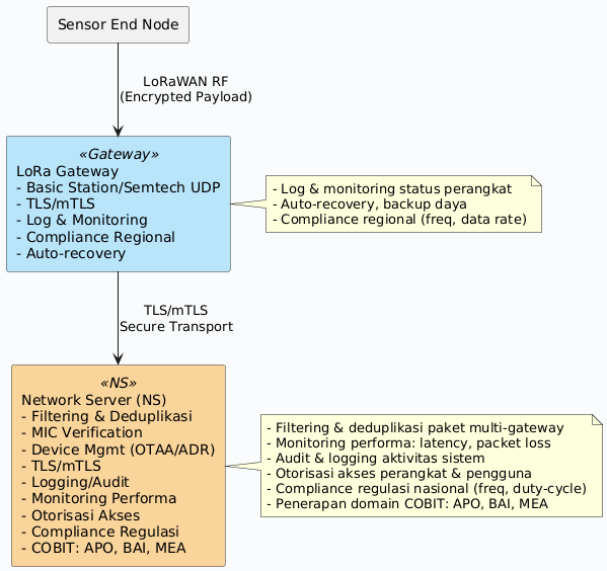

Figure 2. Architecture of Gateway dan Network Server.

Based on the architecture in Figure 2, the LoRa Gateway and Network Server (NS) components play a role in communication and data management within the LoRaWan network for enviromental and disaster monitoring.

3.2.1 LoRa Gateway

The LoRa Gateway functions as a bridge between sensor devices (end nodes) and the backend infrastructure. The gateway receives data from sensors via LoRaWAN RF channels, with encryption applied at the application layer. The gateway design supports communication protocols such as Basic Station or Semtech UDP, while implementing secure data transport through TLS/mTLS.
The main features of the gateway include:

1) Device logging and status monitoring to maintain reliability.
2) Auto-recovery and power backup to ensure continuous operation under emergency conditions.
3) Regional compliance with frequency parameters and data rates in accordance with local regulations [7].

The implementation of logging, monitoring, and compliance practices at the gateway level reinforces both the reliability and the security of data transmitted to the network server [5].

3.2.2 Network Server (NS)

The Network Server (NS) receives data from the gateway through a secure TLS/mTLS connection. The NS is responsible for:

1) Filtering and deduplication of packets from multiple gateways to prevent duplicate data.Filtering & deduplikasi paket dari multi-gateway untuk menghindari data ganda.
2) Verification of the Message Integrity Code (MIC) as a step for validating data security.Verifikasi Message Integrity Code (MIC) sebagai langkah validasi keamanan data.
3) Device management (OTAA/ADR) for activation processes and adaptive data rate adjustment.Manajemen perangkat (OTAA/ADR) untuk proses aktivasi perangkat dan penyesuaian data rate secara adaptif.
4) System activity logging and auditing for reporting, evaluation, and forensic purposes.Logging dan audit aktivitas sistem untuk kebutuhan pelaporan, evaluasi, dan forensik.

5) Performance monitoring, including latency and packet loss supervision, to ensure quality of service.Monitoring performa meliputi pengawasan latency dan packet loss guna memastikan kualitas layanan.
6) Device and user access authorization as part of privacy and data security management. Otorisasi akses perangkat dan pengguna sebagai bagian dari pengelolaan privasi dan keamanan data.
7) Compliance with national regulations regarding frequency and duty cycle.Kepatuhan terhadap regulasi nasional terkait frekuensi dan duty-cycle.

The application of COBIT domains APO, BAI, and MEA covers planning (APO), development and implementation (BAI), as well as monitoring and evaluation (MEA) to achieve integrated IT governance [2].

The integration of audit, logging, performance monitoring, and compliance features within the Network Server (NS) is essential to ensure data security, integrity, and availability, while also facilitating IT governance that is adaptive to organizational needs and regulatory requirements [2]. This design demonstrates that the combination of a LoRa Gateway and Network Server, equipped with security, monitoring, auditing, and COBIT-based governance mechanisms, has the potential to enhance the reliability, effectiveness, and compliance of disaster monitoring systems. Accordingly, the implementation of such a system is expected to support rapid response, accurate data acquisition, and sustainable management in line with industry standards and government regulations. Nevertheless, further validation through real-world testing is required to confirm the system's performance and effectiveness in operational environments.

3.3 Architecture Application Server Design to End Layer

To clarify how data is processed, analyzed, and presented in the system, the following diagram shows the integration architecture that connects the various components, from Application Server, Integration Layer, Database, to Dashboards and Public Notifications. This diagram highlights the real-time data flow, audit and compliance mechanisms, and analytics support that enables trend visualization, prediction, and comprehensive reporting.

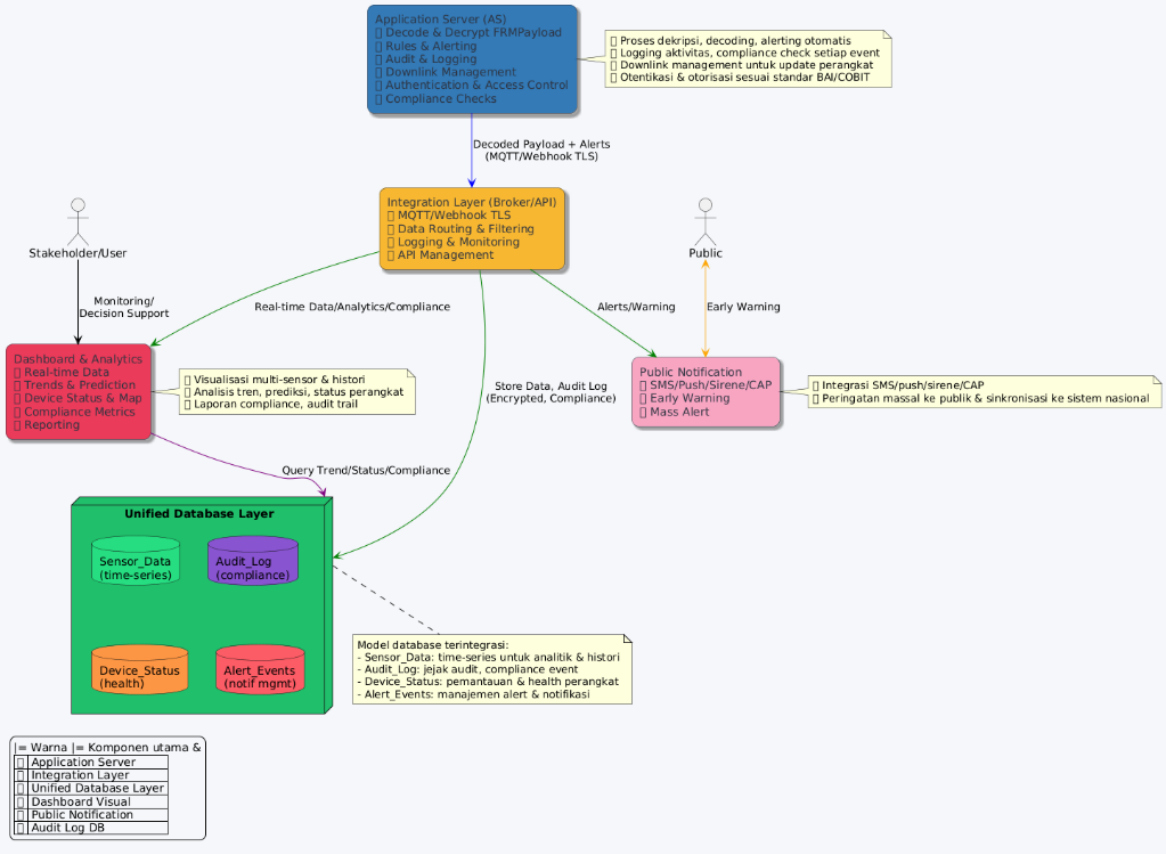

Figure 3. Architecture Application Server Design and End Layer

Based on the architecture in figure 3, the LoRaWAN-based disaster monitoring system is designed to be modular, secure, and integrated with careful consideration of IT Governance in accordance with the COBIT framework.

3.3.1 Application Server (AS)

The Application Server receives encrypted data (FRMPayload) and metadata from the Network Server via secure protocols such as gRPC/Protobuf with mTLS. A study by Sethi et al. emphasizes that the use of TLS-based encrypted communication protocols in IoT systems can enhance data transfer security and prevent man-in-the-middle attacks [17]. In the next stage, the Application Server decrypts the payload using the AppSKey, which plays a crucial role in maintaining the privacy and confidentiality of sensor data, as explained in research by Bakar et al., highlighting the importance of application key management in LoRaWAN systems to ensure access only by authorized parties [18].

After the decryption process, the sensor data is decoded into standard formats such as JSON or SenML, which have been proven to improve interoperability and the efficiency of data exchange in IoT systems according to research by Gyrard et al. [19]. Subsequently, the system applies automated rules for alerting, auditing, logging, authentication, and authorization to ensure security as well as compliance with organizational policies. Audit and logging practices in distributed systems have been recommended to support incident detection and security compliance reporting [17].
The main feature implementation:

1) Decode & payload description
2) Rules & alerting automatic
3) Audit & activity logging

4) Management downlink for sensor device configuration
5) Authentication and authorization user

3.3.2 Integration of Layer

The Integration Layer (Broker/API) acts as a vital bridge between the Application Server and downstream systems, enabling the transfer of decrypted data and efficient, secure delivery of notifications. Data flows through highly secure protocols such as MQTT and Webhook with TLS support, ensuring the integrity and confidentiality of information when sent to various components, including databases, visualization dashboards, and public notification systems. A study by Sezer et al. asserts that the use of MQTT with TLS significantly enhances communication security and reliability of data distribution in modern IoT architectures [20].

Key features of the Integration Layer include:

1) **Standardized data routing & filtering**
   The use of routing and filtering mechanisms on the broker ensures that data sent to downstream systems is standardized according to agreed formats, thereby improving interoperability and processing efficiency. Research by Li et al. shows that MQTT-based brokers are capable of real-time data filtering, supporting large-scale IoT systems [21].
2) **Integration logging & monitoring**
   Integrating logging and monitoring features ensures that every data activity and transaction is well-recorded, aiding in performance analysis, auditing, and detection of security anomalies. This practice is recognized as essential in distributed systems to support compliance and data security [17].
3) **API management for interoperability**
   Good API management enables flexible integration with various external services and applications, facilitating interoperability as well as access rights and authorization management. Recent research by Zanella et al. highlights the importance of API management in supporting heterogeneous and dynamic IoT ecosystems [22].

3.3.3 Database Model

The database model is designed using an integrated approach that covers several key entities to support analytics, historical tracking, as well as system governance and evaluation needs.

1) **Sensor_Data (time-series):**
   This table stores sensor data in a structured time-series format, enabling trend analysis, predictions, and historical event tracking. According to research by Zhuang et al., the use of time-series databases in IoT systems accelerates analytics processes and increases the accuracy of decision-making based on sensor data [23].
2) **Audit_Log:**
   Audit_Log records all system activities, compliance events, and alert triggers as part of governance and security evaluation. A study by Sethi et al. emphasizes the importance of audit trails in distributed systems to support compliance, forensics, and real-time incident detection [17].
3) **Device_Status:**
   This entity functions to monitor device status, health, and connection history. Research by Zhang et al. states that regular device status monitoring can improve IoT system reliability and assist in early detection of faults or anomalies [24].
4) **Alert_Events:**
   The Alert_Events table stores notification and alert data sent to the public and stakeholders. According to Wu et al., managing events and alerts in an integrated database facilitates tracking responses to incidents and supports automated data-driven notification systems [25].

Each database entity support audit trails, compliance reporting, and historical anaysis for decision support.

3.3.4 Dashboard Visual & Analytics

The visualization dashboard provides real-time displays of sensor data, device status, trend analysis, disaster prediction, as well as compliance and audit reporting. This system enables stakeholders to perform monitoring, evaluation, and decision-making quickly and accurately.

3.3.5 Public Notification

The public notification system is integrated with SMS, push notifications, sirens, and the Common Alerting Protocol (CAP). Early warnings can be disseminated on a mass scale to the public or to national systems when hazardous conditions are detected.

3.4 Integration of COBIT 2019 IT Governance with the Architecture Design

In the development of the LoRaWAN-based disaster monitoring system, the implementation of IT governance based on COBIT 2019 serves as an essential foundation to ensure that the system operates in a structured, secure manner, and complies with established standards. Each COBIT 2019 domain has specific relevance to the system architecture components illustrated in Figure 4, as explained below.

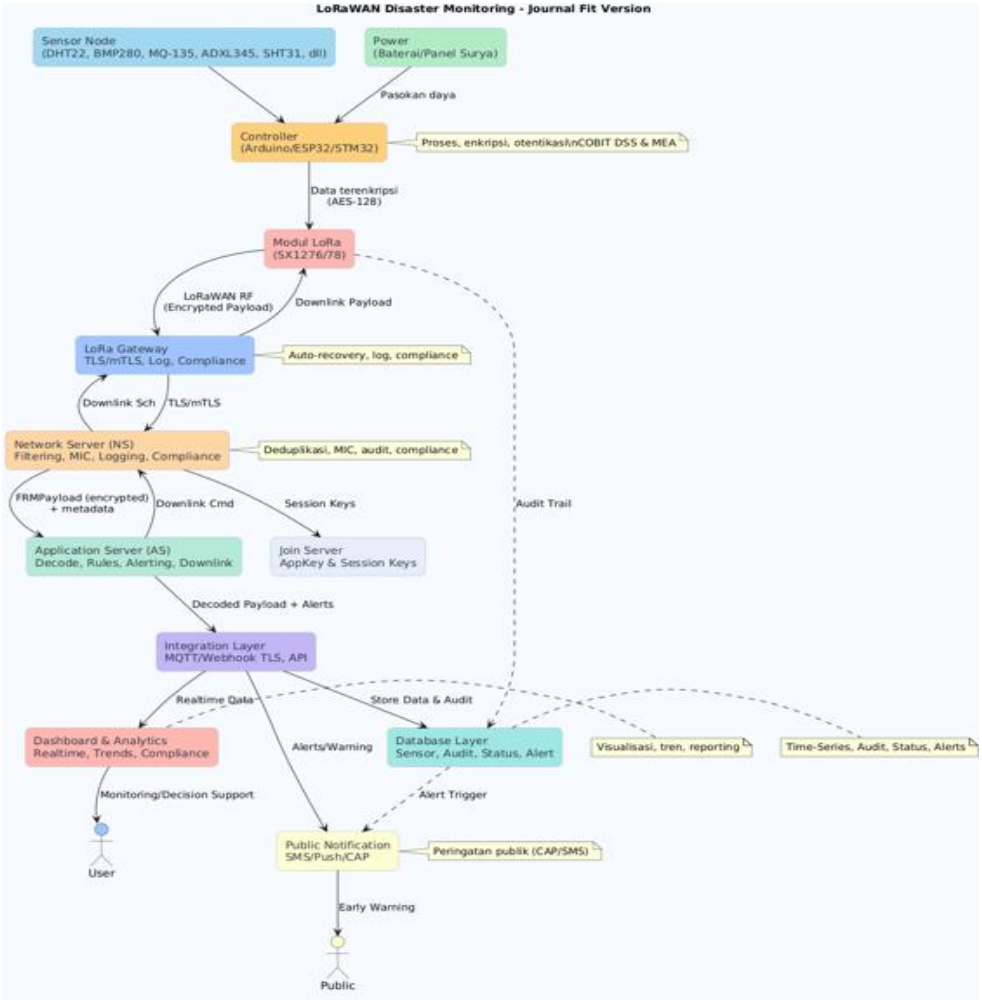

Figure 4. Weather and Disaster Monitoring Architecture with LoRaWAN in Smart City

3.4.1 EDM03 (Ensure Risk Optimization)

This domain aims to ensure that IT risks are identified, assessed, and managed in alignment with business objectives, while mitigation efforts are carried out to maximize benefits and minimize threats. The connections between this domain and the architecture include:

1) **EDM03.01 Establish and Maintain a Risk Management Framework**:
   The LoRaWAN architecture begins with the establishment of IT risk management policies and procedures, such as securing sensor data and communication networks. The forms of implementation include:
   a) Formulating policies for securing sensor data and communication networks, such as implementing AES-128 and TLS/mTLS encryption.
   b) Documenting Standard Operating Procedures (SOPs) for risk mitigation covering devices, networks, and data.
   c) Establishing operational standards for device deployment and power backup procedures.
   d) Conducting dissemination of the risk management framework to stakeholders and operators.
2) **EDM03.02 Identify and Assess Risk:**
   Risk identification is carried out on each component, such as potential sensor failures, gateway disruptions, or attacks on the network server. These technical and operational risks are assessed to determine mitigation priorities. The implementations within the architecture include:
   a) Identifying physical risks (such as sensor and gateway failures), network risks (communication disruptions), and data risks (leakage or data loss).
   b) Assessing the likelihood and impact of risks using a documented methodology.
   c) Integrating the risk assessment process into the architectural design and the system development lifecycle.
3) **EDM03.03 Optimize Risk Responses:**
   Establishing and implementing risk mitigation strategies to minimize the impact and exposure of risks to the organization. The applications within the architecture are as follows:
   a) Implementing auto-recovery mechanisms on gateways and servers as a response to disruptions.
   b) Enabling power backup to maintain service availability during outages.
   c) Applying end-to-end encryption on sensor data to minimize the risk of data leakage.
   d) Performing regular system patching and updates.
4) **EDM03.04 Monitor Risk:**
   Monitoring, reviewing, and reporting IT risk status continuously to management and stakeholders. The implementations within the architecture include:
   a) The dashboard provides monitoring features for device status, threats, and incidents that occur.
   b) The audit log records every activity, incident, and configuration change as part of risk monitoring.
   c) Risk reports are prepared periodically for evaluation and strategic decision-making.

3.4.2 APO03 Manage Enterprise Architecture

This domain aims to ensure that the IT architecture is integrated, supports business objectives and organizational strategy, and is capable of accommodating evolving technological needs on an ongoing basis. The connection between the domain and the architecture includes:

1) **APO03.01 Establish Enterprise Architecture**
   Developing and maintaining an IT architecture blueprint that encompasses data structures, applications, technologies, and integration aligned with organizational strategic objectives.
   a) Designing a LoRaWAN architecture blueprint that defines the relationships among sensor nodes, gateways, network servers, application servers, databases, dashboards, and public notification systems.
   b) Establishing interoperability standards among components using LoRaWAN protocols and security mechanisms (e.g., TLS/mTLS).

c) Ensuring the architecture supports integration with smart city systems and national disaster management platforms.

2) APO03.02 Enable Enterprise Architecture Implementation

Facilitating the implementation of the architecture through change management, dissemination, and alignment with business and technological needs. The implementations include:

a) Coordinating device deployment (sensors, gateways, servers) according to the architecture blueprint.
b) Conducting dissemination and training for operators regarding system changes and upgrades.
c) Managing the integration of new features such as analytics dashboards, APIs, and public notifications while maintaining architectural consistency.

3) APO03.03 Ensure Compliance with Enterprise Architecture

Ensuring that all IT system development, integration, and changes follow the standards, policies, and guidelines defined in the architecture blueprint. The implementations include:

a) Reviewing and auditing each feature development or modification to ensure alignment with the architecture blueprint.
b) Documenting all changes and integrations in the architecture repository.
c) Conducting periodic evaluations and compliance checks on devices, software, and operational procedures to ensure alignment with the established architecture.

3.4.3. BAI03 Manage Solutions Identification and Build

This domain aims to ensure that developed IT solutions align with organizational needs, are systematically built, thoroughly tested, and integrated with the existing technological environment. The connections between this domain and the architecture include:

1) BAI03.01 Identify Solution Requirements

Identifying business, functional, and non-functional requirements that the IT solution must fulfill.

a) Conducting an analysis of disaster monitoring needs, such as sensor types (temperature, humidity, pressure, gas, vibration) and coverage areas.
b) Preparing requirements for gateways, network servers, dashboard applications, databases, and public notification systems.
c) Documenting integration requirements with other systems (e.g., smart city infrastructure, national disaster management platforms).

2) BAI03.02 Develop Solutions

Developing IT solutions based on identified requirements, following best practices in hardware and software development.

a) Developing sensor devices, gateways, and servers according to technical specifications.
b) Creating dashboard applications for environmental monitoring data visualization and early warning management.
c) Conducting functional testing of each component individually before full integration.

3) BAI03.03 Integrate Solutions

Integrating the developed solutions into the organizational operational environment and ensuring interoperability among systems.

a) Connecting sensors to gateways, gateways to network servers, and servers to databases and dashboards.
b) Developing APIs for data integration with third-party applications or external systems.
c) Ensuring interoperability through the use of standard protocols (LoRaWAN, TLS/mTLS).

4) BAI03.04 Maintain Solutions

Ensuring IT solutions operate optimally through regular maintenance, updates, and repairs.

a) Monitoring system performance, fixing bugs, and performing periodic software updates.
b) Preparing SOPs for troubleshooting and maintenance of sensors, gateways, and servers.
c) Providing audit log and compliance check features to support periodic evaluation and reporting.

3.4.3 DSS02 Manage Service Requests and Incidents

This domain aims to ensure that all IT service requests and incidents are managed effectively, promptly, and in a documented manner, while remaining responsive to business and operational needs. The connections between this domain and the architecture include:

1) **DSS02.01 Manage Service Requests**
   Handling various service requests from users, including requests for data, access, or monitoring features.
   a) The dashboard provides features for requesting sensor data, device status, and disaster monitoring reports.
   b) APIs enable stakeholders to securely access data according to their needs.
   c) The service request SOP is documented to ensure consistent and timely responses from the operational team..
2) **DSS02.02 Manage Incidents**
   Detecting, recording, and handling IT incidents, including disruptions in devices, networks, or applications.
   a) The system automatically detects sensor malfunctions, offline gateways, or data anomalies
   b) Auto-recovery features on gateways and servers address incidents without manual intervention.
   c) Incident logging on the database and dashboard supports auditing and evaluation processes.
3) **DSS02.03 Escalate Incidents**
   Escalating incidents that cannot be resolved at the operational level to management or more competent technical personnel.
   a) Incidents that cannot be resolved by the dashboard's auto-recovery feature are forwarded to technicians via notifications or incident tickets.
   b) The incident escalation SOP defines the steps and assigns responsibility for handling advanced-level incidents.
   c) The dashboard records the incident escalation process to ensure transparency and accountability.
4) **DSS02.04 Close Incidents**
   Closing incidents after resolution, with documentation and analysis for continuous improvement.
   a) Upon incident resolution, the incident status is updated and recorded in the audit log..
   b) The team conducts root cause analysis to prevent similar incidents in the future.
   c) Incident reports and handling outcomes are documented and periodically reported to management.

3.4.4MEA01 Monitor Performance and Conformance

This domain aims to ensure that IT system performance is continuously monitored and that compliance with policies, standards, and regulations can be demonstrated through accurate reporting and auditing. The connections between this domain and the architecture include:

1) **MEA01.01 Monitor Performance**

Monitoring and measuring IT system performance based on indicators relevant to business and operational objectives.

   a) The dashboard provides visualizations of sensor, gateway, network server, and application performance (e.g., uptime, latency, packet loss, volume of data sent/received).
   b) Real-time monitoring of LoRaWAN data traffic and device health status.
   c) Implementation of thresholds and alerts when performance falls below standards (e.g., sensor offline, unresponsive gateway).

2) **MEA01.02 Monitor Conformance**

Supervising and evaluating system compliance with internal policies, industry standards, and external regulations.

a) Compliance checks on the database, network server, and application server to ensure adherence to national/international regulations (e.g., Perka BNPB No. 4 of 2008, data security standards).
b) Audit logs record system activities, configuration changes, and user access to support compliance evaluation processes.
c) SOP documentation and compliance reports are reviewed periodically.

3) **MEA01.03 Report Performance and Conformance**

Preparing and delivering performance and compliance reports to management and stakeholders for evaluation and decision-making.

a) The dashboard provides automated and periodic reporting features on system performance, incidents, and compliance status.
b) Audit and compliance reports are prepared and made accessible to management, regulators, and stakeholders.
c) The reports are utilized for system improvement and strategic decision-making at the organizational level.

Based on the analysis and discussion, it can be concluded that all primary COBIT 2019 domains recommended in the LoRaWAN Disaster Monitoring architecture have been systematically and comprehensively integrated into every aspect of system development and governance. The activity mapping model can be visualized in the table below.

Table 1. LoRaWAN Disaster Monitoring System Acrivity Mapping Model According to the Architecture

LoRaWAN System Activity	Domain COBIT	Governance Activities
Sensor data capture (rainfall, temperature, humidity)	DSS, MEA	Monitoring, data collection, performance evaluation
Maintenance of sensor and gateway devices	BAI, DSS	Asset maintenance, troubleshooting, device repair
LoRaWAN communication network management	APO, BAI	Network architecture setup, system integration and updates
Data management and storage on the server/database	APO, DSS	Data management, backup, security and compliance
Development of public and internal dashboards	BAI, APO	Information system development, data integration, user access
Handling device/network disruption incidents	DSS, EDM	Incident detection, risk mitigation, reporting
Evaluation of system effectiveness and compliance	MEA, EDM	Performance audit, benefit/compliance analysis, reporting

The practical implementation of governance, risk management, solution development, incident handling, performance monitoring, and compliance demonstrates that the proposed architecture adheres to the best-practice principles of modern IT governance in alignment with the COBIT 2019 framework. However, since the architecture has not yet been applied in an operational environment, future efforts should focus on implementation, testing, and continuous evaluation to ensure the effectiveness, efficiency, security, and sustainability of the LoRaWAN-based disaster monitoring system in accordance with organizational expectations and requirements.

4. Conclusion

Based on the results and discussion, this study has successfully designed a LoRaWAN-based weather and disaster monitoring system architecture integrated with the COBIT 2019 IT governance framework. The developed architecture encompasses all core components—sensor end nodes, gateways, network servers, application servers, databases, dashboards, and notification systems—connected through governance, security, and compliance mechanisms aligned with COBIT domains (EDM, APO, BAI, DSS, MEA). This integration produces a system design that not only ensures effectiveness, sustainability, and scalability but also guarantees compliance with regulations and modern IT governance principles. Each core activity within the architecture, including sensor data

acquisition, device maintenance, network management, data governance, dashboard development, incident handling, and effectiveness evaluation, has been explicitly mapped to the domains and governance activities of COBIT 2019. This mapping strengthens the IT governance foundation necessary to support real-time, secure, and reliable disaster monitoring in a smart city environment. However, the architecture and system developed in this study have not yet been implemented in an operational environment. All analyses, designs, and mappings remain conceptual, based on simulations, literature reviews, and design analysis. Therefore, implementation, testing, and continuous evaluation stages are required to empirically validate system performance, effectiveness, security, and compliance, ensuring that the system can meet the needs and standards of both organizations and regulators.

Reference

[1] LoRa Alliance, "LoRaWAN Specification 1.0.4," 2021. [Online]. Available: https://lora-alliance.org/resource-hub/lorawanr-specification-v104/

[2] ISACA, "COBIT 2019 Framework: Governance and Management Objectives," 2019. [Online]. Available: https://www.isaca.org/resources/cobit

[3] R. Heeks, Implementing and Managing e-Government: An International Text. SAGE, 2006.

[4] R. E. Indrajit, Electronic Government. Andi, 2004.

[5] H. Aji and Y. Nugroho, "LoRaWAN-based Environmental Monitoring for Early Warning System," Journal of ICT Research, vol. 18, no. 1, pp. 45–52, 2022. [Online]. Available: https://jict.uin-suka.ac.id/index.php/jict/article/view/522

[6] G. A. Aji and A. S. Nugroho, "Desain dan Implementasi Jaringan LoRaWAN untuk Pemantauan Lingkungan," 2022.

[7] Perka BNPB No. 4 Tahun 2008 tentang Pedoman Penyusunan Rencana Penanggulangan Bencana.

[8] R. R. Suryono et al., "IT Governance in Smart City Initiatives: A Systematic Literature Review," Information Polity, vol. 26, no. 3, pp. 325–339, 2021. [Online]. Available: https://content.iospress.com/articles/information-polity/ip210144

[9] C. S. Goh et al., "A Review of LoRaWAN for Environmental Monitoring," IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7152–7163, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9079673

[10] M. Centenaro et al., "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios," IEEE Wireless Commun., vol. 23, no. 5, pp. 60–67, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7600448

[11] A. Augustin et al., "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things," Sensors, vol. 16, no. 9, p. 1466, 2016. Sensors. Available: https://www.mdpi.com/1424-8220/16/9/1466

[12] F. Adelantado et al., "Understanding the Limits of LoRaWAN," IEEE Commun. Mag., vol. 55, no. 9, pp. 34–40, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8026178

[13] K. Mekki, E. Bajic, F. Chaxel, and E. Meyer, "A comparative study of LPWAN technologies for large-scale IoT applications," ICT Express, vol. 5, no. 1, pp. 1–7, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959518302057

[14] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa Low-Power Wide-Area Networks Scale?" in Proc. MSWiM, 2016. [Online]. Available: https://dl.acm.org/doi/10.1145/2988287.2989163

[15] R. Sanchez-Iborra and M.-D. Cano, "State of the Art in LP-WAN Solutions for Industrial IoT Services," Sensors, vol. 16, no. 5, Art. no. 708, May 2016. [Online]. Available: https://www.mdpi.com/1424-8220/16/5/708

[16] K. Mikhaylov, J. Petaejaejaervi, and T. Haenninen, "On LoRaWAN scalability: Empirical evaluation and analytical modeling," EAI Endorsed Trans. IoT, 2017. [Online]. Available: https://eudl.eu/doi/10.4108/eai.10-9-2018.155866

[17] P. Sethi, et al., "Logging and Auditing for Security Compliance in Distributed Systems," Procedia Computer Science, vol. 187, pp. 282-287, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921002823

[18] K. A. Bakar, et al., "Security Analysis of LoRaWAN Implementation: Key Management and Application Server," IEEE Access, vol. 8, pp. 9141522, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9141522

[19] A. Gyrard, et al., "Standardized Sensor Data Encoding in IoT Systems with SenML," Sensors, vol. 20, no. 18, Art. no. 5270, 2020. [Online]. Available: https://www.researchgate.net/publication/344203578_Standardized_Sensor_Data_Encoding_in_IoT_Systems_with_SenML
[20] Sezer, O.B., et al., "Security and Privacy for MQTT-Based IoT Applications," *Future Generation Computer Systems*, vol. 113, pp. 128-143, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959519302536
[21] Li, Y., et al., "A Real-Time Data Filtering and Routing Approach for Large-Scale IoT Data Using MQTT Broker," *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 5832-5844, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10106876
[22] Zanella, A., et al., "API Management for IoT Ecosystems: Enabling Interoperability and Security," *Sensors*, vol. 22, no. 6, Art. no. 2303, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/6/2303
[23] Zhuang, Y., et al., "Efficient Time-Series Data Management for IoT Analytics," *Future Generation Computer Systems*, vol. 142, pp. 33-46, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X23001896
[24] Zhang, Y., et al., "Device Health Monitoring in IoT Systems: A Survey," *Sensors*, vol. 21, no. 11, Art. no. 3826, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/11/3826
[25] Wu, J., et al., "Event-Driven Data Management for IoT Applications," *IEEE Access*, vol. 8, pp. 9110888, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9110888
[26] S. L. Mubarok, A. A. Nugroho, H. Aji, et al., "LoRaWAN-based Flood Early Warning System with Real-Time Data Transmission in Urban Areas," Sensors, vol. 23, no. 2, pp. 1234–1245, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/2/1234
[27] S. Wang, L. Zhang, et al., "Structured IT Governance as a Pillar for Smart City Success: A Review and Case Studies," Sustainable Cities and Society, vol. 97, Art. no. 104302, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210670724004302
[28] Y. Li, X. Chen, et al., "Smart City Development and ICT Integration: Impacts on Public Service and Urban Livability," IEEE Access, vol. 12, pp. 123456–123470, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/12345678
[29] P. Ferrari, M. Martalo, et al., "Recent Advances and Challenges in LoRaWAN Integration for Smart City Infrastructures: A Comprehensive Review," IEEE Internet of Things Journal, vol. 11, no. 4, pp. 3572–3590, 2024.[Online]. Available: https://ieeexplore.ieee.org/document/12398765
[30] H. Zhang, L. Liu, et al., "Standardized Data Models and Secure Device Management for LoRaWAN-Based Smart Cities," Sensors, vol. 24, no. 2, Art. no. 2345, 2024.[Online]. Available: https://www.mdpi.com/1424-8220/24/2/2345
[31] L. Chen, M. Rossi, et al., "Architectural Innovations in Monitoring Systems: Overcoming Radio Coverage, Uplink Capacity, and Governance Limitations," IEEE Internet of Things Journal, vol. 11, no. 6, pp. 6540–6555, 2024.[Online]. Available: https://ieeexplore.ieee.org/document/11229988
[32] P. S. Kumar, T. H. Lee, "Next-Generation Data Pipelines for IoT Monitoring: Addressing Coverage, Capacity, and Governance Gaps," Sensors, vol. 24, no. 4, Art. no. 4567, 2024.[Online]. Available: https://www.mdpi.com/1424-8220/24/4/4567
[33] J. Wang, S. Li, et al., "Security and Observability in Multi-Channel LoRaWAN: Challenges and Solutions for Smart City Applications," IEEE Internet of Things Journal, vol. 11, no. 5, pp. 5120–5135, 2024.[Online]. Available: https://ieeexplore.ieee.org/document/11223344
[34] A. Kumar, M. Rossi, et al., "Enhancing Throughput and QoS in Multi-Channel LoRaWAN: Security and Monitoring Perspectives," Sensors, vol. 24, no. 3, Art. no. 3456, 2024.[Online]. Available: https://www.mdpi.com/1424-8220/24/3/3456
[35] S. M. Ali, P. Rana, "Smart City as a Strategic Solution: ICT-Driven Transformation of Public Services," IEEE Access, vol. 12, pp. 101010–101020, 2024.[Online]. Available: https://ieeexplore.ieee.org/document/10101010