Analysis of System Level Design Implementation in High-Level Design and Application Architecture on Support Stream System

Yosua Alvin Adi Soetrisno[1], Eko Handoyo[1], Imam Santoso[1], Hafiz Alfian Aqsha[2]

[1] *Lecturer, Department of Electrical Engineering, Diponegoro University, Central Java, Indonesia*
[2] *Student, Department of Electrical Engineering, Diponegoro University, Central Java, Indonesia*

ABSTRACT

The research was conducted at PT. PLN Persero is Indonesia's largest electricity company providing electricity services to approximately 60 million customers nationwide. PLN Persero manages the information systems that support the company's business operations. One of the challenges faced by the division is the presence of scattered local unit applications throughout Indonesia, which are difficult to control and update centrally. Therefore, there is a need to transition from local unit applications to centralized applications in the Support Stream by integrating the applications through Enterprise Architecture Planning (EAP). This study employed a case study method, collecting data through interviews, observations, and document analysis to analyze High-Level Design architecture and application integration in Central PLN. The research findings indicate that implementing System Level Design at the High-Level Design (HLD) and Application Architecture levels can ensure architectural consistency and quality and reduce future application maintenance costs. Thus, implementing System Level Design at the HLD and Application Architecture levels is crucial in application development as it can provide significant long-term benefits to the organization.

Keywords: *Application Architecture, Enterprise Architecture Planning, System Level Design, High-Level Design, Support Stream, Application Integration*

1. INTRODUCTION

PT Perusahaan Listrik Negara (PLN) is a state-owned enterprise (BUMN) engaged in electricity generation, transmission, and distribution and is one of the significant electricity sellers in Indonesia [1]. The central administration of PLN's business operations throughout Indonesia is managed by the PLN Headquarters located in Jakarta. The headquarters consists of several departments and divisions assisting business processes. The Information Technology Systems Division has a Support Stream encompassing various units, including customer service, power generation, human capital, distribution, projects, finance, transmission, procurement & supply chain, support, and infrastructure. Among these streams, this discussion focuses on the Support Stream because it involves numerous support applications developed by each unit that has not yet been integrated with the central system.

Among the many units that develop applications, some functions overlap with those developed at the central level. This overlap complicates and slows PLN's business processes and increases future application maintenance costs. Therefore, a solution and innovation are required to integrate these support applications into a centralized platform that facilitates PLN's business processes. The solution is implementing High-Level Design (HLD) and Enterprise Architecture Planning (EAP) for PLN's application system.

The inventory results show that there are 76 local unit support applications. Numerous applications challenge the efficiency of both central and local operations since they still rely on different applications. Additionally, the central office cannot monitor the performance of each unit effectively. This issue can be resolved by integrating the unit applications into a centralized system, allowing for better monitoring and efficient data management for each unit. Before integrating the applications, it is essential to design the architecture of the Support Application System to ensure that all functions and features are accommodated and interconnected properly, enabling both units and the central office to efficiently utilize the Centralized Support Application according to their specific needs.

The purpose of conducting this research is to gain knowledge about the job scope, work domains, and competencies involved in working within the system of an Information Technology company. It aims to expand insights into the structured web development process within the company, understand the design aspects of website entities, and design a web project using Figma.

The design of the Application System Architecture utilizes the High-Level Design Architecture method to facilitate programmers and system management in comprehending the interconnections between applications, resulting in a centralized application. Implementing HLD and EAP in PLN's application system can aid in developing consistent and high-quality applications. This HLD ensures the alignment of application systems within the organization and reduces future application maintenance costs. Additionally, this implementation enhances the efficiency and effectiveness of PLN's business processes. The gaps between local or unit applications and the central application can be identified using the HLD method, encompassing functionality, scope, and integration. This analysis helps understand the current AS-IS application condition and determine the desired To-Be changes. Consequently, it facilitates the design of the integrated application architecture [2].

The problems encountered in this research revolve around analyzing the feature gaps between local and central support applications, performing integration mapping from local applications to the central application, and designing an application system architecture that all units can utilize. In this study, the author focuses on specific issues related to analyzing the usability of both local unit and central applications in their support functions, analyzing the features of each application in the support stream, mapping local unit applications to the centralized application, and analyzing the features that need to be developed, added, or removed in those applications.

2. THEORETICAL FOUNDATION

2.1 Application Architecture

Application architecture focuses on designing an application, determining the placement of components that form a system, and how they communicate. Computer architecture aims to define the primary application types and the data required to process and support enterprise business functions. In application architecture, it is essential to identify and document the critical requirements for processing data and supporting business operations comprehensively [3].

2.2 Enterprise Architecture Planning

Enterprise Architecture Planning (EAP) is an approach developed by Steven H. Spewak to construct enterprise architecture driven by data and business requirements [4]. EAP involves defining the architecture for utilizing information to support business operations and establishing plans for implementing that architecture [5].

In Enterprise Architecture Planning (EAP), there are four layers to determine and plan the implementation of information system architecture: Initiation Planning, Current Enterprise State Assessment, Future Enterprise Plan Review, and Plan Development.

2.3 System Level Design

System-level design is the process of designing an information system as a whole. The stages of system-level design include high- and low-level design, implementation, integration and testing, deployment, and relaunch. System Level Design aims to input system specifications and requirements and generate hardware and software models, which are then synthesized [6].

2.4 Sparx System Enterprise Architecture

Sparx Systems Enterprise Architect is a visual design and modeling tool based on OMG UML. This platform can be used in software system construction and design, business process modeling, and industry-based domain modeling. Companies and organizations use it to model their system architectures and implement those models in application lifecycle management (ALM).

System modeling with UML provides a foundation for modeling various aspects of organizational architecture and the ability to create a basis for planning and implementing new or modifying existing systems. This type of modeling can encompass aspects ranging from organizational or system architecture layout, business process re-engineering (BPR), business analysis and service-oriented architectures, and web modeling [7] to application and database design, as well as re-engineering and development of embedded systems [8].

In addition to system modeling, Enterprise Architect also encompasses important points in application lifecycle management (ALM), from requirements management to the design, construction, testing, and maintenance phases, as well as supporting traceability, project management, and change management of these processes. For example, it provides tools for developing model-based application code using an internal integrated development platform.

The user base ranges from programmers and business analysts to enterprise architects within organizations, from small development companies, multinational corporations, and government organizations to international industry standard bodies [9]. Enterprise Architect was initially released in 2000 by Sparx Systems. The product was initially created as a UML modeling tool for UML 1.1 modeling, but it has evolved to include other OMG UML specifications such as 1.3, 2.0, 2.1, 2.3, 2.1, and 2.5.

3. RESULTS AND DISCUSSIONS

3.1 Needs Analysis

Based on the mentioned issues, which involve the development of applications by multiple units, PT PLN has ended up with numerous applications not integrated with the centralized application. To reduce production and application development costs and improve the support applications' performance, effectiveness, and efficiency, implementing System Level Design and Application Architecture is necessary to facilitate the integration of local or unit applications with the centralized application.

This implementation aims to map applications precisely. Several steps can be taken for application integration, starting from the required application features for each unit and aiming to facilitate the integration process. System Level Design, using High-Level Design, helps understand the current state (as-is) of the support application system and the integrated state (to be). Before integration, it is essential to identify the features of each unit's support applications and the features required by the units so that they can be accommodated in the centralized support application. Using the EAP method in the Application Architecture components makes it easier to map the features of unit and centralized applications.

Based on the compiled analysis, this Application System Architecture has several gradual outputs, such as mapping support applications, Support Application Feature Matrix/Relation, High-Level Design of Support Applications, and Integration of Local Support Applications into the centralized system.

3.2 Application Information Data Filtering

The objective of this stage is to obtain information regarding the applications and features existing in the support applications of each unit. In this case, data or information from the support applications is filtered through interview methods. Directly conducting interviews with the application's Business Process Owner (BPO) in each unit makes it easier to filter application data. The required information includes the purpose of creating the application, the features used, the currently being developed, the planned features to be developed, and any application constraints to support the integration needs of local applications into the centralized system.

In those mentioned above, it can be observed that interviews with the Application BPOs of each unit from Sumatra to Papua were conducted via Zoom meetings. The results of the interviews include information related to the applications, such as the existing features and the required features that have not been accommodated. It is evident from the above that information filtering is stored using an Excel form. By utilizing the features in Microsoft Excel, extracting the required information for creating application mappings and identifying feature gaps between local and centralized applications becomes easier. This Excel table facilitates a more structured approach to developing application architecture and system-level design.

3.3 Support System Architecture

In Enterprise Architecture Planning, the application architecture stage is situated in the Future Plan Review layer of the company. The application architecture's purpose is to define the necessary applications for managing data and supporting the existing business functions within the company. After obtaining data or information from interviews, these data will be processed and categorized into relevant categories within the Supportive Applications. The Application Portfolio Catalogue modeling and the Feature Relationship Matrix will be utilized among the various methods available.

The Application Portfolio Catalog stage aims to define a comprehensive list of supportive applications the company utilizes across its various units. Supportive Applications within PLN refer to the applications that assist the company in managing its operations more effectively and efficiently, enhancing customer service, and improving corporate performance.

Fig- 1 Application Portfolio Catalogue for Support System

Based on the interview data, there are 28 Supportive Applications developed by several units. These applications are categorized into seven groups that support the operational activities of the central company. The categorization of supportive applications can be seen in Figure 1. For more detailed information regarding the seven categories in the supportive applications, they are explained as follows.

1. Occupational Health and Safety (K3L)

This application focuses on Occupational Health and Safety (K3) aspects. This research aims to assess and evaluate the use of information technology concerning unsafe conditions, unsafe actions, near misses, accidents, and work-related illnesses.

2. Audit

We need an application that supports the audit and risk assessment processes.

3. Risk Management

This application is involved in risk profiling, risk calculations, and risk mapping. It helps the company identify and map existing risks to minimize or improve the situation.

4. Helpdesk

We require an application that supports reporting and monitoring incidents or disruptions.

5. General Administration (GA) Services

An application is designed to facilitate employee access to services such as transportation and meeting room reservations.

6. Fixed Assets

This application manages the inventory and monitoring of company assets.

7. Corporate Performance

An application that consolidates and centralizes the performance of each unit into corporate performance data.

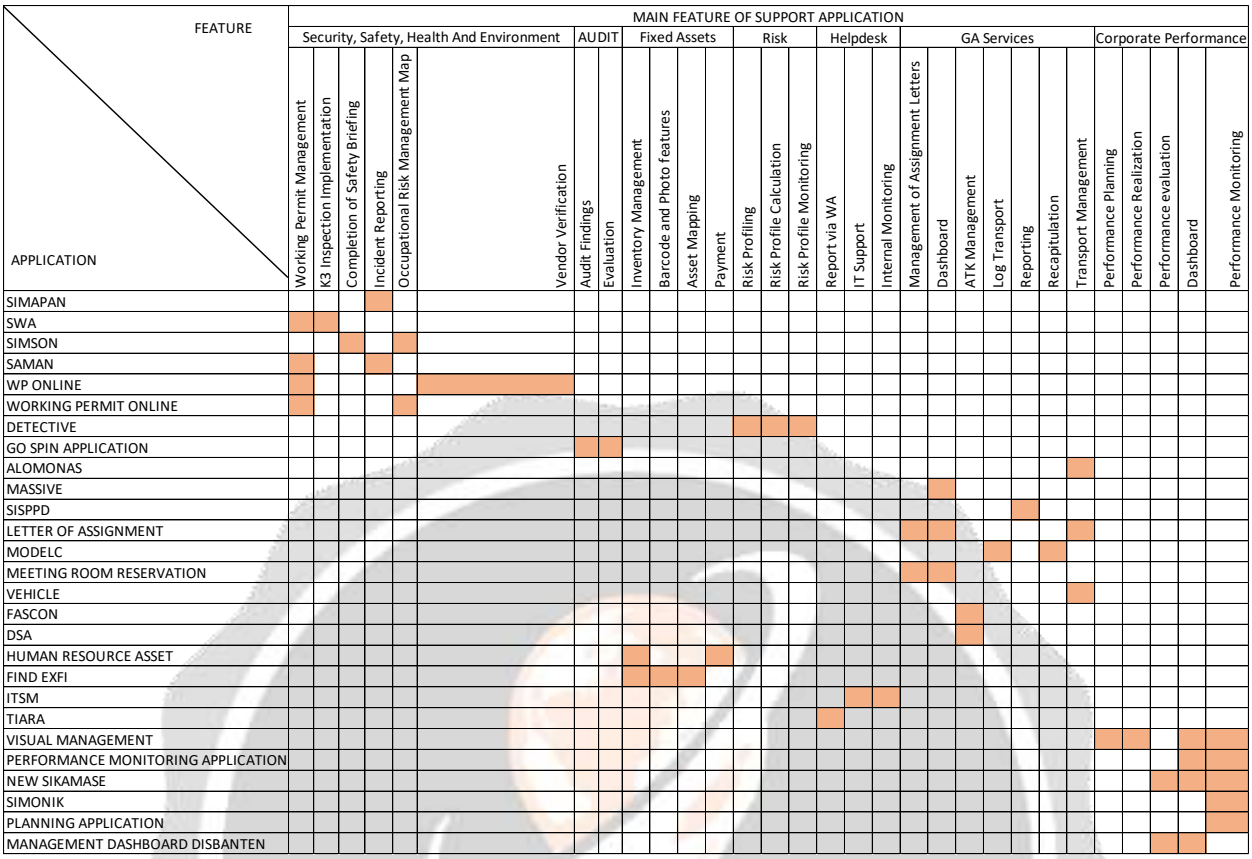

Fig- 2 Relational Matrix Organized by Supporting Applications

After the supporting applications have been grouped according to their categories, a feature application relation matrix, as shown in Figure 2, is required to facilitate the integration process of the applications.

The feature-application relation matrix stage aims to identify the business functions directly supported or performed by the supporting applications. Generally, the mapping of the relationship between supporting applications and business functions is done as follows:

1. The assignment of application usage is based on the organizational business functions' needs, which are determined by reviewing the business processes and data utilized.
2. The policies and available services within the applications support the business functions.
3. Gaps were analyzed between the application's roles in supporting the organizational business processes.
4. The application's roles were determined in supporting business functions and identifying the requirements for future application changes.
5. Feature gaps were identified in both decentralized and centralized supporting applications.

3.4 High-Level Design Application Architecture (Information System) Support

In the High-Level Design (HLD) stage of the supporting application architecture, all the categorized and processed information regarding the application features will be formed into a System-Level Design with HLD levels to facilitate the determination of the integration process of the supporting applications in each category. The purpose of HLD is to describe the application system, database design, services, platforms, and the inter-module relationships containing the features from the created matrix.

HLD has two parts or application architecture flows: the As-Is and To-Be conditions. The As-Is condition depicts the current state of the application architecture being used. In contrast, the To-Be condition represents the flow that illustrates the adjusted or updated state of the supporting application architecture, also known as the future application architecture state. Each part includes features or application flows, the application names themselves, and they will be mapped according to the features listed in the matrix from several applications. Utilizing the

Application Portfolio Catalog and the Feature Application Relation Matrix will facilitate the process of creating the Application Architecture with System-Level Design at the High-Level Design (HLD) level.

As mentioned earlier, the K3L (HSE) supporting the application is related to occupational safety and health. The application's High-Level Design (HLD) can be depicted based on the features present in each unit, as shown in the diagram below.

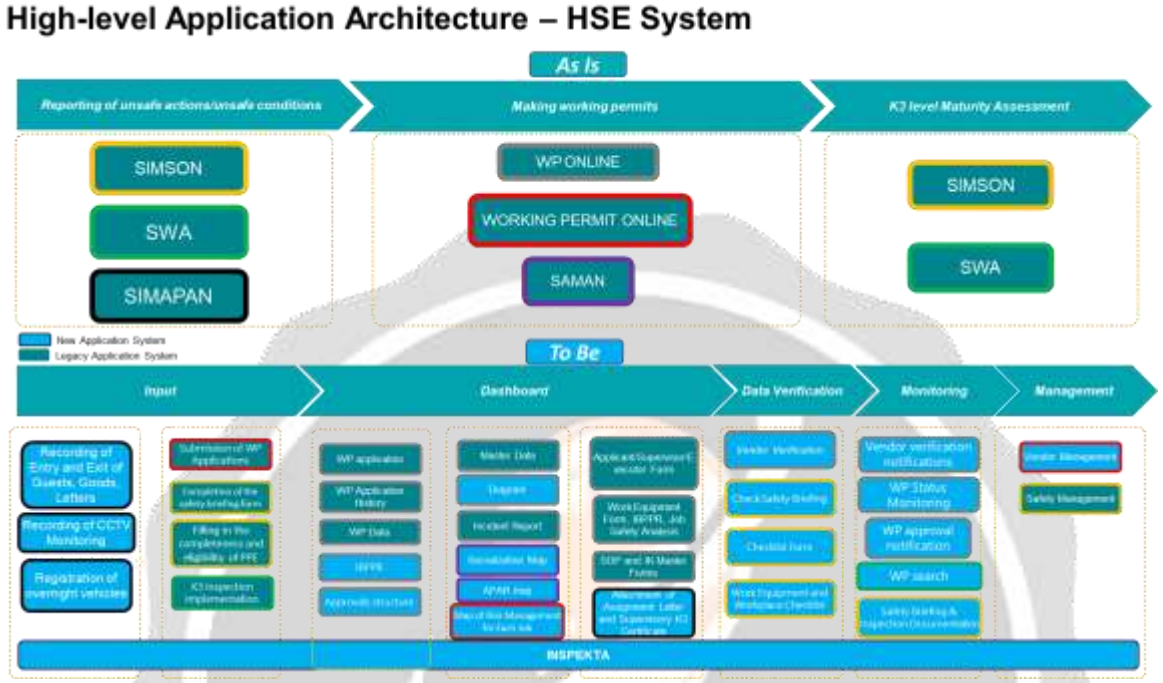

Fig- 3 HLD HSE Support Application

Based on Figure 3, two conditions depict the architecture of the K3L supporting the application. Firstly, the As-Is condition represents the current state of the application architecture. This stage illustrates how the current architecture of the applications is functioning. Multiple applications may support each flow or feature, while some may not be present in specific applications.

Secondly, the To-Be condition represents the future state of the K3L supporting application architecture after it has been updated. Compared to the As-Is condition, several applications already possess the main features required for the K3L-supporting application. However, some applications lack certain essential features for the centralized K3L-supporting application. Creating or merging these features into a single centralized K3L supporting application system is possible could address this. The main features of this application include Working Permit (WP) creation, reporting unsafe actions/conditions, and K3 Maturity Level assessment. Additionally, there are additional features used by several units, including the following:

1. Recording of Incoming/Outgoing Goods, Guests, and Letters
2. Recording of Overnight Vehicle Stays
3. Diagrams
4. Fire Extinguisher (APAR) Map
5. Socialization Map
6. Risk Management Map for Each Job
7. Attachment of Assignment Letters and K3 Supervisor Certificates
8. Data Verification with Notification Features
9. WP Monitoring (Search, WP Notification, Safety Briefing, and Inspection Documentation)
10. Vendor Management

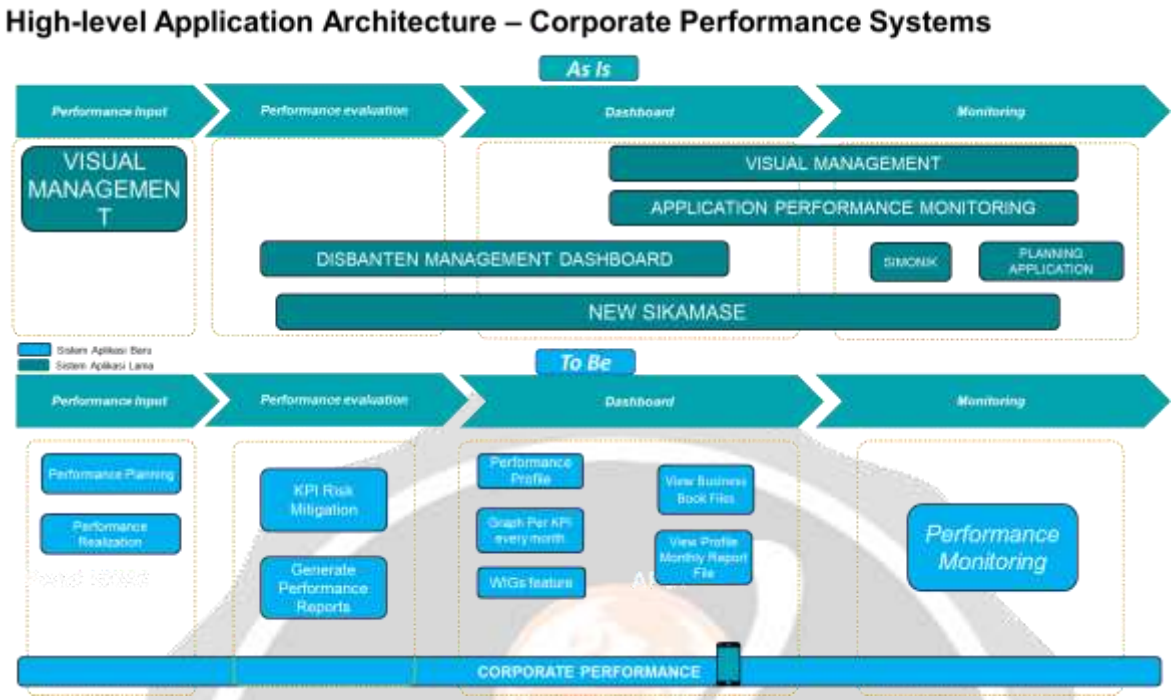

Fig- 4 HLD Corporate Performance Support Application

Figure 4 shows two conditions describing the application architecture supporting Corporate Performance. The first is the As-Is condition or current condition. At this stage, it describes the current state of the application architecture. From each current flow or feature, several applications still use either the same or different features.

The two To-be conditions or future conditions describe the updated Corporate Performance supporting application architecture condition. If viewed from the as-is condition, several applications already have the main features of the Corporate Performance supporting application. This Corporate Performance supporting application is still only available in a few units. But with this application, it will make it easier to make a performance at the company. So that from several corporate performance-supporting applications, the units will be combined into one centralized corporate performance-supporting application. The application's main features are Performance Input, Performance Evaluation, Dashboard, and Performance Monitoring. The following are the contents of the main features.

1. Performance Planning Input
2. Performance Realization Input
3. KPI Risk and Mitigation
4. Generate Performance Reports
5. Performance Profile
6. Graph Per KPI every month
7. WIG feature
8. View Business Book Files
9. View Profile Monthly Report File
10. Performance Monitoring

The Audit Supporting Application is designed to assist in audit processes and risk assessments. The application's High-Level Design (HLD) can be depicted based on the features present in each unit, as shown in the diagram below.

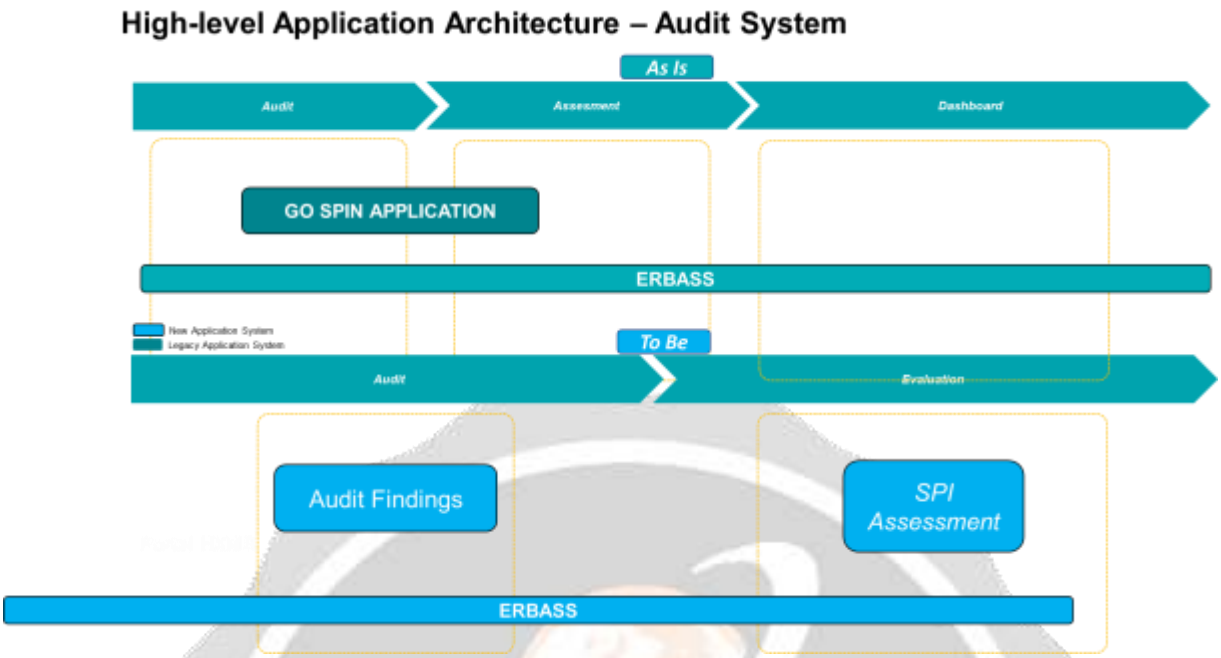

Fig- 5 HLD Audit Support Application

Based on Figure 5, two conditions depict the architecture of the Audit Supporting Application. The first condition is the As-Is condition, which represents the current state of the application architecture. In this condition, it is evident that only one unit application possesses the Audit and Assessment system feature, namely the GO SPIN application. Additionally, there is a centralized audit-supporting application called ERBASS.

The second condition is the To-Be condition, which represents the future state of the Audit Supporting Application architecture after it has been updated. When comparing it to the As-Is condition, the GO SPIN application already has the main features required for the Audit Supporting Application. However, regarding functionality, the GO SPIN application is more closely related to compliance functions, while ERBASS is more operational. Therefore, it may be inappropriate or less suitable to include ERBASS within the Audit Supporting Application.

The Risk Supporting Application is designed to facilitate risk profiling, risk calculation, and risk mapping. This application assists companies in understanding and mapping existing risks, enabling them to minimize or even mitigate those risks effectively. The application's High-Level Design (HLD) can be depicted based on the features present in both unit and central levels, as shown in the diagram below.

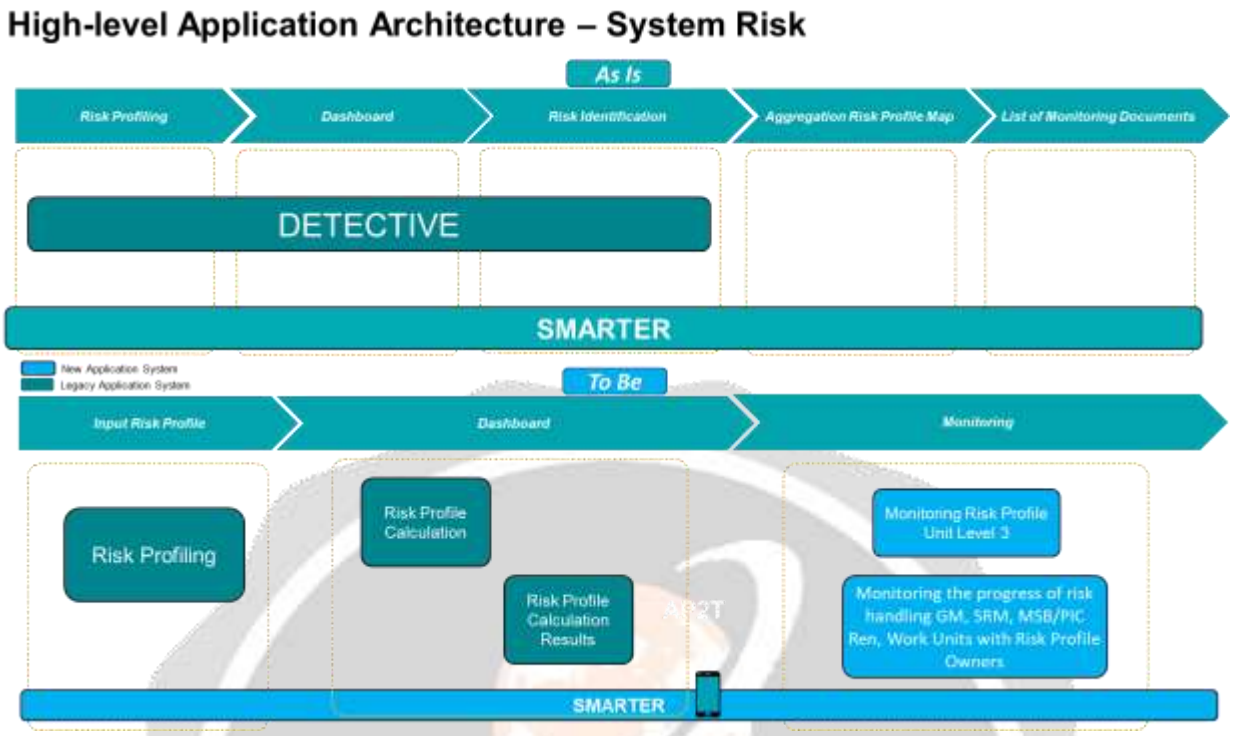

Fig- 6 HLD Risk Support Application

Based on Figure 6, two conditions depict the architecture of the Risk Supporting Application. The first condition is the As-Is condition, which represents the current state of the application architecture. In this condition, it is evident that only one unit application, namely DETEKTIVE, possesses the features of Risk Profile Creation, Dashboard, and Risk Identification. Additionally, there is a centralized risk-supporting application called SMARTER.

The second condition is the To-Be condition, which represents the future state of the Risk Supporting Application architecture after it has been updated. Compared to the As-Is condition, the DETEKTIF application already has the main features required for the Risk Supporting Application. However, a monitoring feature is missing in SMARTER's centralized risk-supporting application. Therefore, the features from each application will be merged into one centralized application.

1. Risk Profile Creation
2. Risk Profile Calculation
3. Risk Profile Calculation Results
4. Unit Level 3 Risk Profile Monitoring
5. Risk Management Progress Monitoring for GM, SRM, MSB/PIC Ren, Risk Profile Owner Work Unit

The Fixed Asset Management Support Application is an application that handles the inventory and monitoring of required assets. The Asset Management Support Application features in each unit can be depicted in the Application Architecture's High-Level Design (HLD), as shown in the image below.

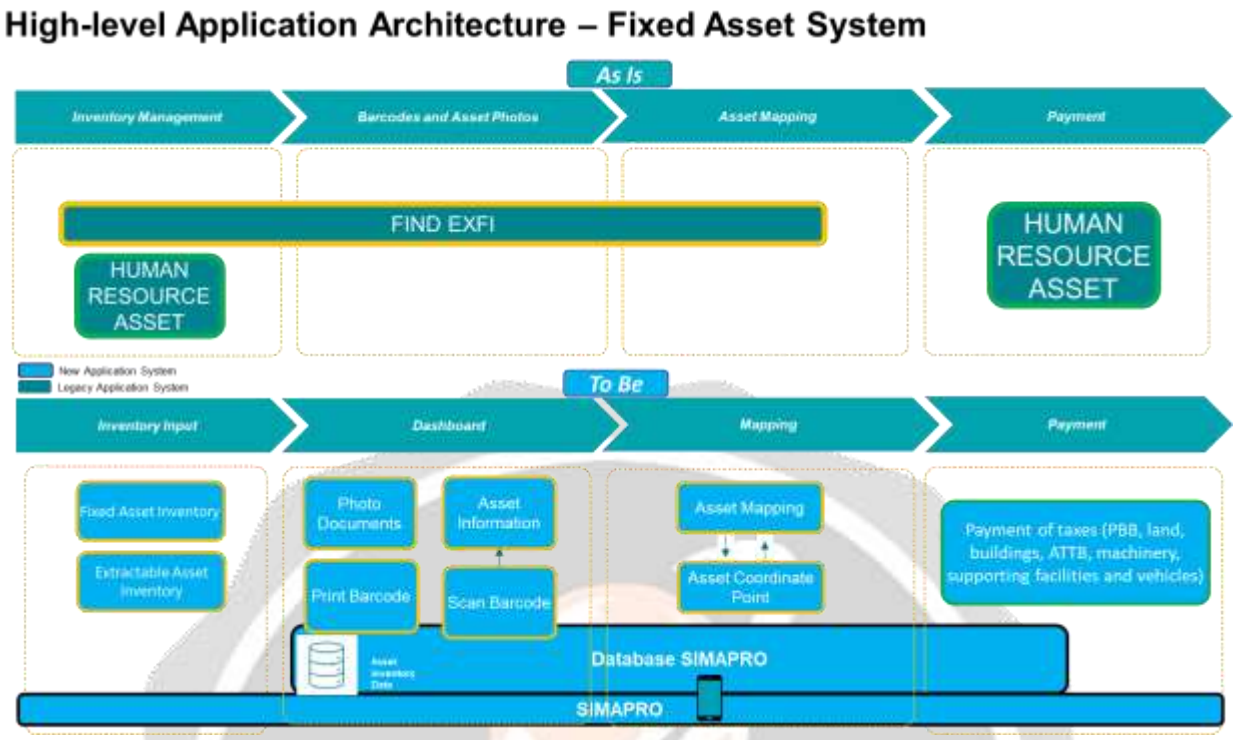

Fig- 7 HLD Fixed Asset Support Application

Figure 7 shows two conditions that depict the architecture of the Fixed Asset Management Support Application. The first condition is the As-Is condition, which represents the current architecture of the application. In this stage, several applications still use similar or different features. Only two applications from 2 units are involved.

The second condition is the To-Be condition, which represents the future state of the Fixed architecture of the Asset Management Support Application. Considering the As-Is condition, several applications already have the main features of the Fixed Asset Management Support Application. These two applications can be combined into a centralized application. The application's main features include Inventory Input, Dashboard, Mapping, and Payment. The fixed asset data will also be stored in the application's database. The details of these main features are as follows.

1. Fixed Asset Inventory
2. Non-Compilable Asset Inventory
3. Document Photos
4. Barcode Printing
5. Asset Information
6. Barcode Scanning
7. Asset Mapping
8. Asset Coordinates
9. Tax Payments (Property Tax, Land, Buildings, Equipment, Supporting Facilities, and Vehicles)

The Helpdesk support application is an application that provides support for handling reported issues and incidents, as well as monitoring their resolution. The High-Level Design (HLD) Architecture of the Helpdesk support application, depicting its features in each unit, can be illustrated in the diagram below.

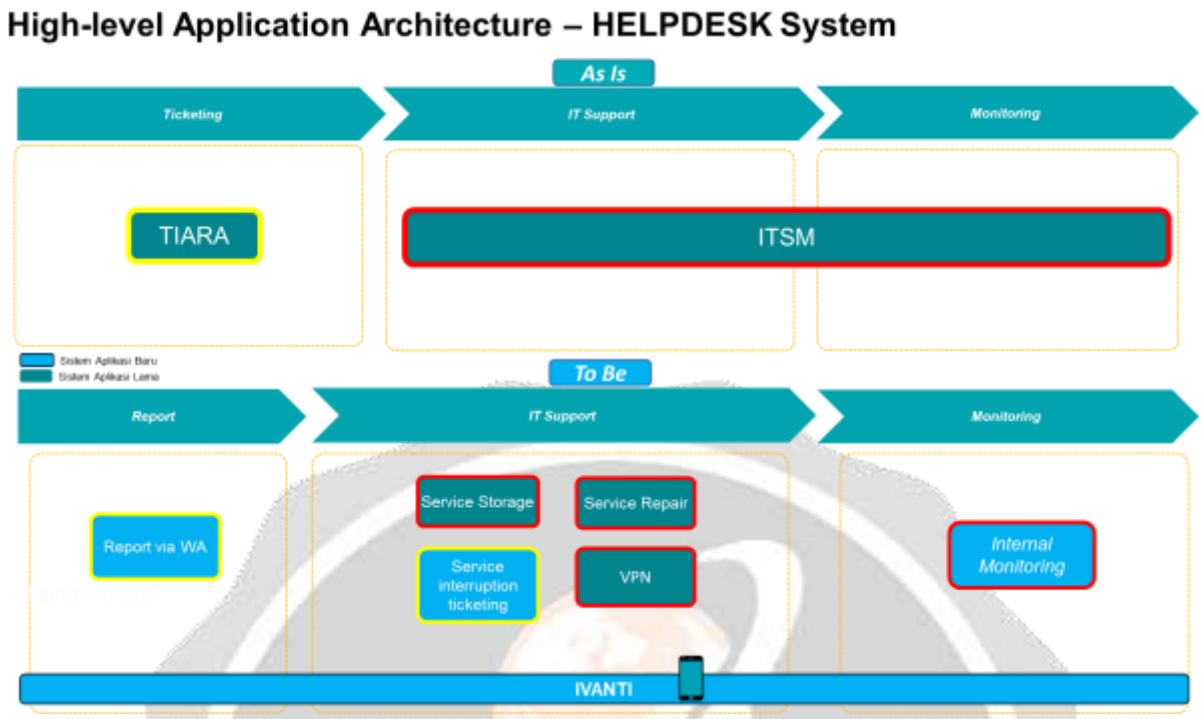

Fig- 8 Helpdesk Support Application HLD

Figure 8 shows two conditions that depict the architecture of the Helpdesk support application. The first condition is the As-Is condition, which represents the current state of the application's architecture. In this condition, several applications are used across different units, some of which have similar features while others have different ones. There are two local Helpdesk support applications, namely TIARA from the KALTIMRA unit and ITSM from the SULUTENGGO unit.

The second condition is the To-Be condition, which represents the updated architecture of the Helpdesk support application. Upon reviewing the As-Is condition, it is evident that several applications already have the main features of a Helpdesk support application. These applications can be consolidated into a centralized application. The main features added from the local Helpdesk support applications include Ticketing, IT Support, and Monitoring. Here are the details of these main features.

1. Report via WhatsApp: This feature allows users to report issues or problems through the WhatsApp messaging platform. It provides a convenient and user-friendly channel for users to submit their reports.
2. Service Storage: This feature involves storing and organizing service-related data. It ensures that all relevant information, such as user details, reported issues, and service history, is securely stored and easily accessible for reference and analysis.
3. Service Ticketing: This feature enables creating and managing service tickets for reported incidents or service requests. It includes functionalities such as ticket creation, assignment to the appropriate support personnel, tracking the status of tickets, and providing updates to users regarding their ticket status.
4. Service Resolution: This feature focuses on resolving reported issues or fulfilling service requests. It includes troubleshooting, problem analysis, implementing fixes or solutions, and verifying that the issue has been resolved satisfactorily.
5. VPN (Virtual Private Network): This feature provides a secure and encrypted connection for users accessing the Helpdesk application remotely. It ensures that sensitive data and communications are protected from unauthorized access, especially when users connect to the application from external networks or locations.
6. Internal Monitoring: This feature involves monitoring the performance and availability of the Helpdesk application internally. It includes tracking key performance indicators, system health checks, resource utilization monitoring, and generating reports or alerts to identify potential issues or bottlenecks within the application infrastructure.

The GA Services Support Application is designed to facilitate various services related to transportation booking, meeting room reservations, and other similar tasks, providing convenience and accessibility to employees. The

application's High-Level Design (HLD) Architecture, as depicted in the image below, outlines the features available in each unit.

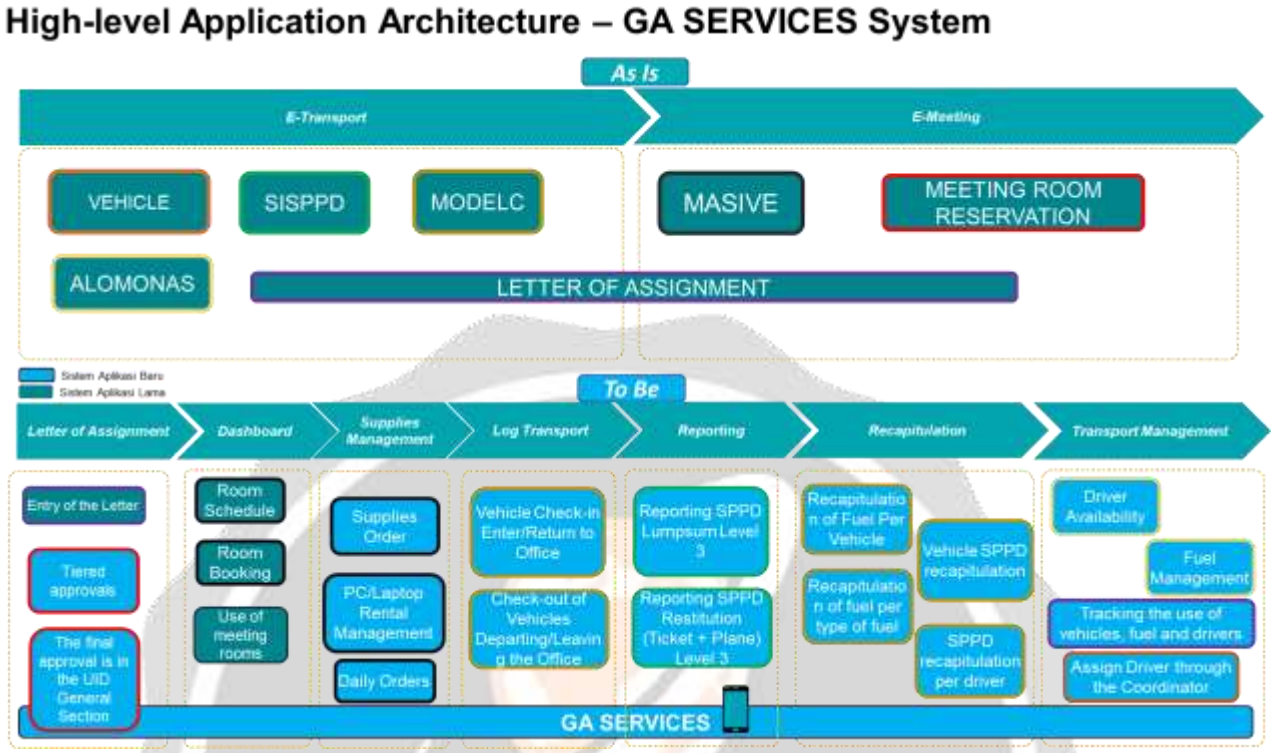

Fig- 9 HLD Application Support GA Services

Figure 9 shows two conditions that depict the architecture of the GA Services Support Application. The first condition is the As-Is or current state, which represents the current working architecture of the application. In this condition, several applications utilize both similar and different features. The current state includes two main features of the GA Services application: E-Transport and E-Meeting.

The second condition is the To-Be or future state, which illustrates the updated architecture of the GA Services Support Application. Upon reviewing the current state, several applications already possess the main features required for GA Services support. These applications can be consolidated into a centralized application. Additionally, the GA Services Support Application unit has incorporated new main features, such as Letter of Assignment, Dashboard, ATK Management, Log Transport, Reporting, Recapitulation, and Transport Management. The following are details of these main features.

1. Letter of Assignment Entry
2. Hierarchical Approval
3. Final Approval at General Affairs Department UID
4. Room Schedule
5. Room Booking
6. Meeting Room Usage
7. Stationery Ordering
8. PC/Laptop Rental Management
9. Daily Booking
10. Check-in for Vehicles entering/returning to the office
11. Check-out for Vehicles departing/leaving the office
12. Lumpsum SPPD Reporting Level 3
13. Restitution SPPD Reporting (Ticket + Flight) Level 3
14. Fuel Recapitulation per Vehicle
15. Fuel Recapitulation per Fuel Type
16. SPPD Recapitulation per Vehicle
17. SPPD Recapitulation per Driver
18. Driver Availability

19. Fuel Management
20. Vehicle, Fuel, and Driver Usage Tracking
21. Driver Assignment through Coordinator

3.5 Support Application Integration

After conducting an inventory, it is known that PT. PLN has 318 unit applications that cover various functions such as customer service, distribution, transmission, generation, projects, procurement & supply chain, human capital, finance, support, and infrastructure. PT. PLN plans to integrate local unit and centralized applications and perform integration mapping based on specific treatment criteria. Table 1 shows the action of integration mapping of each application.

Table -1: Application Integration Mapping

Criteria	***Action***	**Information**
App features are already in the centralized app.	***Switch to Centralized Applications***	The unit application is deactivated and transferred to a centralized application.
Application functionality already exists in a centralized application, but several features have not been accommodated in a centralized application.	***Switch to Centralized Applications (with Enhancement)***	Enhancement is carried out on the centralized application, after which the unit application is deactivated and transferred to the centralized application.
Application functionality does not yet exist in a centralized application but is used by many units.	***Switch to Centralized Applications (Create New)***	A new centralized application is developed, after which the unit application is deactivated and transferred to a centralized application.
Application functionality does not yet exist in a centralized application and is only used in specific units, and does not store confidential data such as customer or company data.	***Keep Local***	Unit applications can still be operationalized without integration with a centralized application and with due observance of established security standards, such as the pentest process, etc. Centralized applications or DWH are allowed to retrieve data from unit applications when needed.
A unit application that functions for monitoring local network networks.		
The functional unit application is connected to OT equipment.		
The application is obsolete and is no longer relevant to run due to changes in business processes and business needs	***Switch Off***	Unit application deactivation is performed.

In the research process, the authors only focus on supporting application streams. Based on interview data processed using the EAP Application Architecture Stage method, starting from grouping application categories using the Application Portfolio Catalog and Application Feature Relations Matrix to determine the gap between the unit and central applications. Then a High-Level Design is made to make it easier to read clearly and in detail in application integration. When viewed from the Supporting Application HLD system, it can be concluded that all Supporting Applications can be integrated into 1 Centralized Supporting Application according to their respective categories.

Table -2: Support Application Integration Mapping

Support Application Category	***Switch to a Centralized Application (with Enhancement)***	***Switch to a Centralized Application (Create New)***	**Centralized Support Application Name**
Security, Safety, Health, and Environment	SAMAN, SIMAPAN, SIMSON, Working Permit Online, SWA, WP Online		INSPEKTA
Audit	GO SPIN Application	APLIKASI GO SPIN	ERBASS
Risk	Detective		SMARTER
Fixed Asset	Human Resource Asset, FIND EXFI		SIMAPRO
GA Services	ALOMONAS, "MASIVE", SISPPD, Letter of Assignment, MODELC, Meeting Room Reservation, Vehicle, FASCON, DSA		GA SERVICES (E-Meeting, E-Transport, E-SPPD, E-ATK)
Helpdesk	TIARA, HELPME, ITSM STG		IVANTI
Corporate Performance		Visual Management, SIMONIK, NEW SIKAMASE, Performance Monitoring Application, Planning Application, Disbanten Management Dashboard	Corporate Performance

Table 2 shows it can be seen that there are 28 Unit Support Applications from 7 Functions or Categories. From the number of applications, they will be integrated according to the indicators in the Application Integration Mapping table. It can be concluded that there are only two appropriate actions for the Support Applications using these indicators. Out of the 28 Unit Support Applications, the integration will be done by switching to the Enhanced Centralized Application and creating a new Centralized Application, resulting in 7 Centralized Support Applications.

The HSE category has 6 Unit Support Applications: SAMAN, SIMSON, SIMAPAN, WP Online, Working Permit Online, and SWA. They will be integrated into the Enhanced Centralized Support Application called INSPEKTA. Integrating these many applications into the centralized application may pose some challenges.

In the Audit category, there is only one Unit Support Application, GO SPIN, which will be switched to the Centralized Support Application, ERBASS, either with enhancement or by creating a new application. Since there is only one application from the unit, integrating it into the centralized application should be relatively more straightforward. However, considering GO SPIN and "ERBASS" different functionalities, they can be combined into a new centralized application or kept as a local one.

Similarly, under the Risk category, only one Unit Support Application can be easily integrated into the centralized application with enhancement.

In the Fixed Assets category, 2 Unit Support Applications will be integrated into the Centralized Support Application with enhancement. Integrating these applications should be relatively straightforward in terms of features and compatibility.

GA Services has 4 Function Support Applications: E-Meeting, E-Transport, E-SPPD, and E-ATK. Out of the 9 Unit Support Applications, each application can be integrated according to its function. For example, Reservation of Meeting Rooms and Massive can be integrated into E-meetings. "MODELC", ALOMONAS, Vehicles, Assignment Letters, and SISPPD can be integrated into E-Transport. FASCON and DSA can be integrated into E-ATK. Considering the existing compatibility, integrating these applications into the centralized application with enhancement would be of medium difficulty.

Under the Helpdesk category, 3 Unit Support Applications will be integrated into a single Centralized Support Application called IVANTI. Considering the challenges of these applications, it can be concluded that integrating them into the centralized application with enhancement should not be too difficult.

In the Corporate Performance category, 6 Unit Support Applications will be integrated into a new Centralized Support Application called Corporate Performance. Considering the challenges of these applications, it can be concluded that creating a new centralized application for integration should not be too difficult.

4. CONCLUSIONS

Based on the research conducted at PT PLN (Head Office) regarding the implementation of a centralized application system's high-level design architecture, the following conclusions can be drawn:

Based on the analysis of feature gaps between local and centralized support applications, the following features need to be added to the Centralized Application: INSPEKTA (WP Monitoring), E-Transport (Vehicle, Driver, and Fuel Tracking), Corporate Performance (Performance Monitoring), ERBASS (Audit Findings), SMARTER (Risk Identification), SIMAPRO (Asset Barcode and Photo), and Ivanti (IT Support Ticketing).

Based on the High-Level Design and Enterprise Architecture Planning, it is concluded that there are 28 Local Support Applications that will be integrated into 7 Centralized Support Applications, and they depict the menu features that need to be added to the Centralized Support Applications.

Based on the Application Integration Mapping, 22 Local Support Applications will be integrated into Centralized Support Applications with feature enhancements, and 6 Local Support Applications will be integrated into a new Centralized Support Application called Corporate Performance.

6. REFERENCES

[1] Profil Perusahaan. PLN. 2021. 7 Mei 2023. https://web.pln.co.id/tentang-kami/profil-perusahaan

[2] Krisetya, N. (2014). Penerapan Enterprise Architecture Planning (EAP) pada Pembuatan Arsitektur Data, Aplikasi dan Teknologi: studi Kasus PT. Sumber Sehat (Doctoral dissertation, Program Studi Sistem Informasi FTI-UKSW)

[3] Betri, T. J., Utami, E., & Al Fatta, H. (2017). Perancangan Arsitektur Aplikasi Learning Management System Di Universitas Slamet Riyadi. IJAI (Indonesian Journal of Applied Informatics), 2(1), 17-32.

[4] EAP: Spewak, S. H., & Hill, S. C. (1993). Enterprise architecture planning: developing a blueprint for data, applications and technology. QED Information Sciences, Inc.

[5] Surendro, K. (2007). Pemanfaatan Enterprise Architecture Planning untuk perencanaan strategis sistem informasi. Jurnal Informatika, 8(1), 1-9.

[6] System Level Design: Pop, P., Eles, P., & Peng, Z. (2004). Analysis and synthesis of distributed real-time embedded systems. Springer Science & Business Media.

[7] Truyen, F. (2010). Enacting the Service Oriented Modeling Framework™(SOMF™) using Enterprise Architect.

[8] Rosenberg, D., & Mancerella, S. (2010). Embedded systems development using SysML: an illustrated example using enterprise architect. Sparx Systems Pty Ltd and ICONIX, 4-14.

[9] Cox, S. (2009). GEOSCIENCE AUSTRALIA AND CSIRO: Developing the GeoSciML interoperability standard with Enterprise Architect.