International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 7 Issue 5, May - 2023, Pages: 10-15

Analysis of High Level Design Application Centralized Application System Architecture in Stream Distribution

Ajub Ajulian ZM, Enda Wista Sinuraya, Yuli Christyono, Bambang Winardi

Department of Electrical Engineering, Diponegoro University, Semarang, Indonesia
e-mail: bbwinar@gmail.com

Abstract— *PT. PLN is a company engaged in providing electricity in Indonesia. In industry 4.0, PT. PLN uses information technology to support the company's business needs. PT PLN has units spread throughout Indonesia where each local unit also builds its own application. Based on the results of the 2022 inventory, there are 344 active unit applications. In order to improve the efficiency and productivity of the company, PT. PLN performs Enterprise Application Integration on the unit's local application. Enterprise Application Integration has several stages that need to be passed, one of which is application architecture. Application architecture is important because it is the basis for application design including features in the application which will support the business process. Application Architecture has several stages in software design, namely System Level Design, High Level Design, and Low Level Design. This report is specifically to discuss the application of High Level Design in the IT Planning and Strategy subdivision of the Information Technology Systems Division of PT. PLN.*

Keywords: Enterprise Application Integration. Application Architecture. System Level Design, High Level Design. Low Level Design.

1. INTRODUCTION

PT PLN is a company engaged in the energy sector which has 11 subsidiaries that support the company's performance and services. Because PT PLN is a company whose business is engaged in the energy sector that provides services to all regions in Indonesia, PT PLN has many branches for electricity distribution. Within each local unit, PT PLN creates applications that support the needs of that unit. Examples of needs supported by local unit applications such as customer service, generators, human capital, distribution, projects, finance, transmission, procurement & supply chain, support, and infrastructure.
This becomes ineffective because many applications are made by local units, making it difficult to integrate at the center. Based on the results of unit application inventory, there are 318 local unit applications. This problem can be solved by doing architectural planning for local application units.
There are several treatment criteria that can be given to local unit applications, including (1) Keeping the application in the local, (2) Keeping the application in the local but making improvements, (3) Multiplying to a centralized application, (4) Switching to the application centralized but making improvements, and (5) Switching to a centralized application but creating a new application. The method used to carry out this analysis is by interviewing each application regarding features, strengths, and weaknesses so as to obtain data to support decisions in the treatment to be carried out on the application. To achieve application integration in PLN, it is necessary to do mapping which will be carried out on the central PLN application. Furthermore, it is necessary to use the High Level To Be Plan to Construct method. This method will show High Level Design including functional, scope, and integration.

2. SOFTWARE DESIGN

Software process design involves creating a framework that enables the software development team to plan, execute, and monitor the steps involved in building the software. A software process is a series of interrelated activities that transform requirements into a software product. Three stages . The main elements in the software design process are:

1. SLD (System Level Design) or system level design: this stage involves designing the general system architecture. At this stage, the software designer considers the overall system objectives, decides on the technology to be used, determines the data flow, and develops a system test plan. SLD focuses more on overall design and does not pay attention to implementation details.
2. HLD (High-Level Design) or high-level design: this stage involves designing the system in more detail. At this stage, the software designer makes detailed specifications regarding functionality and interactions between components in the system. HLD is more focused on a more detailed and specific design compared to SLD
3. LLD (Low-Level Design) or low-level design this stage involves designing each system component and module in detail At this stage, the software designer creates an implementation plan for each component and hunts down technical design details, such as algorithms and data structures which will be used. LLD focuses more on the implementation and details of each component in the system. These three stages are interrelated and are usually carried out sequentially in the software design process. SLD stages assist software designers in planning the overall system architecture, while HLD assists in designing detailed functionality and

interactions between components in the system. LLD assists in designing the technical implementation of each component and module in the system.

2.1 System Level Design

System Level Design System level design in applications is a stage in application development that aims to design the overall system architecture. This process includes planning from the technical, architectural and infrastructure aspects needed to run applications efficiently and effectively. Here are some steps that need to be taken in carrying out a system level design on an application:

1. Determine system requirements. First of all, it is necessary to determine the system requirements you want to design. This includes the functionality of the application, the type of data to be processed, the number of users who will use the application, and so on.
2. Designing the application architecture. After determining system requirements, the next step is to design the overall application architecture. This includes determining the required components, such as databases, servers, and network infrastructure.
3. Create a data flow diagram. Data flow diagrams will help in visualizing how data flows through the application. This helps in ensuring that data is processed correctly and according to system requirements.
4. Determine the application infrastructure. After designing the application architecture and data flow diagrams, the next step is to determine the required application infrastructure. This includes system selection
   operations, servers, databases, and technology used.
5. Determine final development and testing. Need to define a development and test plan for the application. This includes creating a development schedule, defining test methods and designing an effective testing strategy.

2.2 High Level Design

High Level Design (HLD) is the high level design stage of a system or application to be built. At this stage, designers focus on understanding and defining functional and non-functional requirements, system architecture, data flows, user interfaces, and other key features.

This stage is very important because it will define the overall system architecture, including the appearance design and system features, as well as the technology to be used. The following are some steps that can be taken in the HLSD stage:

1. Analyze functional and non-functional requirements. At this step, the designer must understand the functional and non-functional requirements of the system or application to be built. Functional requirements describe what the system should do, while non-functional requirements describe how the system should perform, such as security, speed, and scalability.
2. Make an overview of the system architecture.
   After understanding the requirements, the designer must create an overview of the system architecture. This involves identifying the main components and how they will interact with each other to form the system. This overview of the system architecture will serve as the primary guide during system development.
3. Create data flow diagrams or use case diagrams Designers must create data flow diagrams or use case diagrams to visually explain how the system works Data flow diagrams show how data flows through the system, while use case diagrams identify user interactions with the system.
4. Planning the user interface. The user interface should be well thought out at this stage. Designers must consider the features that users need and create easy-to-use and intuitive interfaces.
5. Determine the features and technology to be used. The designer must determine the features and technologies to be used to build the system. This involves selecting programming languages, databases, and other technologies.
6. Determine the technical specifications and division of tasks. Finally, the designer must determine the technical specifications needed to build the system, as well as divide the tasks between the development team members. This involves determining the size of the team, the development schedule, and the responsibilities of each team member.

3. Discussion

3.1 System Requirements

Based on the results of the inventory carried out, PT. PLN has 318 application units consisting of customer service, distribution, transmission, generator, project, procurement & supply chain, human capital, finance, support and infrastructure functions. PT. PLN will integrate local unit applications with centralized applications.

Based on the results of the application identification, from 318 local unit applications that have the same business process, they can be integrated into 33 centralized applications with details:

1. Switch to centralized applications, as many as 24 local unit applications can be recommended to switch to existing centralized applications, namely PLN WIKI, A2MRT, BBM Oniline, Lisdes Application, PMO.
2. Switch to centralized applications & enhance, as many as 146 local unit applications can be recommended to switch to centralized applications by enhancing features, namely the GA Services, INSPECTA, CRM, IVANTI, SMARTER, New

Srintam, New ITO, AGO, Dashboard Iventori, EAM applications Transmission (Legacy), MAXIMO (EAM Distribution), HXMS Portal, E-Procurement, E-Contract, E-KHS, FIX, E-Budget, Vendor Invoicing Portal (VIP), ERBASS, SIMAPRO

3. Switch to centralized applications & create new, as many as 48 local unit applications can be recommended to switch to new centralized applications namely Corporate Performance Applications, SCADA Portals, Centralized E-AIL, VMS (Vendor Management System)
4. Keep local, there are still 100 Unit Applications which are mostly used for local network monitoring (Operational STI), information/news portal for each Unit, Power Plant Monitoring Application, 4DX Application, Customer Satisfaction Survey Application, Internal Document Management Application, Application Risk, Audit Application.
5. Switch Off, as many as 26 unit applications. In the process of practical work, the author only focuses on distribution streams. Local application integration planning for distribution stream units, using local unit application architecture targeting. The targeting results can be seen in the image below.

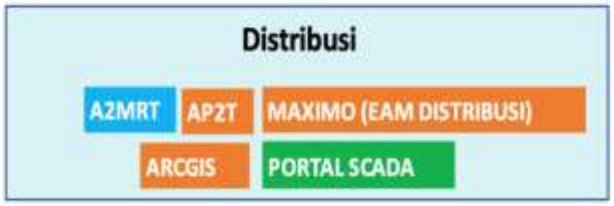

Figure 4.1 Targeting Integration of Local Applications Distribution Stream Units

In the picture above it can be seen that the application on the distribution stream will be integrated into 5 centralized applications, namely A2MRT, AP2T, MAXIMO (EAM DISTRIBUTION), ARCGIS, and SCADA PORTAL.
In the Distribution stream, there are many probis that are fulfilled by local unit applications to support the needs in Distribution. Data for probis and the local application of the probis support unit can be seen in the image below.

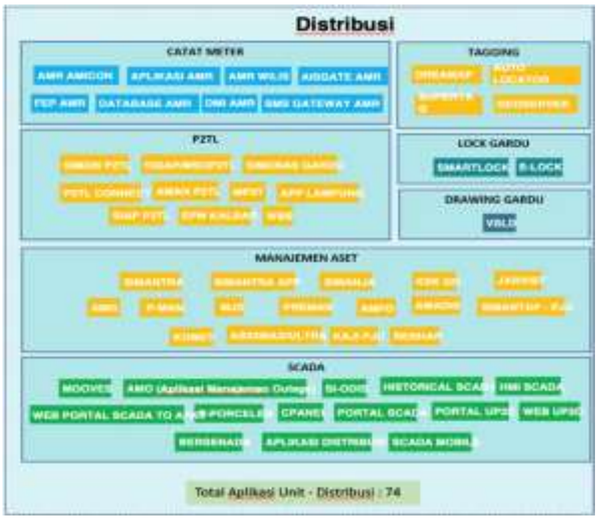

Figure 4.2 Distribution Stream Application Mapping

As can be seen in the figure, the distribution stream has 6 functions, namely meter recording, tagging, P2TL, Substation Lock, Substation Drawing, Asset Management, and Scada. The Distribution Stream has 74 local unit applications that support their functional requirements and probis. The author focuses on conducting a High Level Design analysis on the treatment criteria for Switch to Existing Centralized Applications and Enhance Features and the criteria for Switching to Existing Centralized Applications and Create New Centralized Applications which are centered on several centralized applications, namely AP2T, MAXIMO (EAM DISTRIBUTION), ARCGIS, and SCADA PORTALS.

3.2 Integrasi Aplikasi AP2T

The AP2T application is used for P2Tl (Controlling Electric Power Consumption). Applications that will be integrated with the AP2T application are AP2T (ACEH Unit), SIAP P2TL (Banten Unit), Simon P2TL (Riau Unit), Commercial Application (Central Java

Unit), SIGAP/MSOP2TL (Central Java Unit), AMOREST (East Kalimantan Unit), AMAN P2TL (East Kalimantan Unit), APP Lampung (Lampung Unit), SMES (Jakarta 1 Unit), and AMOREST (South Sumatra). Based on the interview results, the features that are not available in the centralized AP2T Application are obtained below.

1. Monitoring P2TL Operational Targets
2. Monitoring the Determination of P2TL
3. Monitoring the Progress of TO Inspection
4. Call monitoring (2,3)
5. Warning monitoring (1,2,3)
6. Monitoring Unloading Complete
7. Input and monitoring of P2TL data which has been checked
8. Monitoring of P2TL letters (to find out how many warning letters have been sent to P2TL customers)
9. Upload P2TL Customer Operation Targets
10. Upload P2TL Customer Findings (Normal or violation)
11. Upload Medium Voltage Customer Findings
12. Upload the post-P2TL customer files
13. Legal Products related to P2TL
14. Monitoring Post P2TL Customer Status
15. APP Validation (for AMR meters)
16. Change Meters
17. Maintenance of AMR
18. Monitoring Yantek (APP interference)
19. Monitoring the Use of Seals
20. Monitoring of New Post Restitution
21. Presentation of the new pair restitution database
22. Shrink per UP3 (There is a display of losses per substation)
23. Analysis of substation losses
24. Monitoring of Meter Anomaly Anomaly (Dashboard, APKT Interference Report, Realization of Meter Anomaly, Upload Anomaly)
25. Monitoring of LBKB (Read Code Repeat Report)
26. End Voltage Monitoring
27. Big Data Masters
28. Utilities
29. Realtime P2TL Finding Map (versi Android)

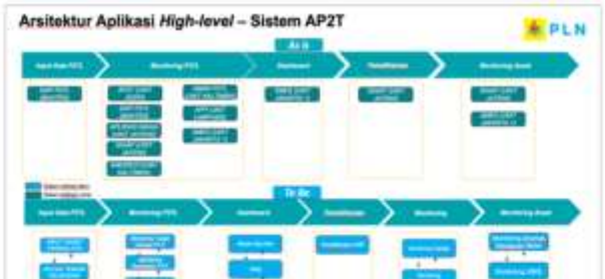

Figure 4.3 As Is and To Be Analysis of the AP2T System

The above is the result of As-Is and To Be analysis on the AP2T application system where in the As Is condition for the P2TL data input feature only owned by SIAP P2TL (Banten Unit). Then for the P2TL MONITORING feature it is available in the AP2T application (Aceh Unit), SIAP P2TL (Banten), NIAGA APPLICATION (Central Java Unit), SIGAP (Central Java Unit), AMORES (East Kalimantan Unit), AMAN P2TL (East Kalimantan Unit), APP (Lampung Unit)), SMES (Jakarta Unit 1). Furthermore, the dashboard function is only owned by the SMES application (Unit Jakarta 1). The maintenance function is provided by SIGAP (Central Java Unit). The Asset Monitoring function is provided by SIGAP (Central Java Unit) and SMES (Jakarta 1 Unit). In the To Be condition, the system is centralized in the AP2T application. Where the P2TL Data Input function already provides P2TL operation target input features, Upload customer findings, and Upload TM customer findings. The P2TL Monitoring function provides features for monitoring P2TL operational targets, monitoring P2TL implementation, monitoring TO inspection progress, monitoring calls and warnings, and monitoring completed unloading. The dashboard function provides master big data features, utilities, and realtime P2TL finding maps. The maintenance function provides AMR maintenance features. The monitoring function provides monitoring features, monitoring the use of seals, and monitoring the restitution of new pairs. In asset monitoring, there are features for monitoring meter disturbance anomalies, monitoring LBKB, and monitoring end voltages.

3.3 MAXIMO Application Integration (EAM DISTRIBUTION)

Aplikasi MAXIMO (EAM DISTRIBUSI) merupakan. Aplikasi yang akan melakukan integrasi ke aplikasi MAXIMO (EAM DISTRIBUSI) yaitu MJD (unit Banten), RENHAR (Unit Sumut), P-MAN (Unit Kaltimra), JARVIST (Unit Kaltimra), AMADIS/AMPD (Unit Sumbar), SIMANTRA (Unit PPB), dan ASIIXMAX (Unit Jatim 1).

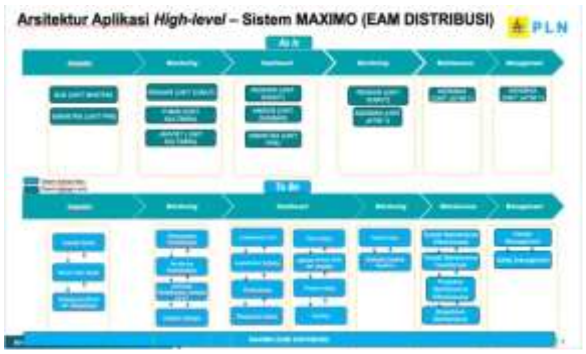
Arsitektur Aplikasi High-level – Sistem MAXIMO (EAM DISTRIBUSI)
PLN
As Is
To Be

Figure 4.4 As Is and To Be Analysis of the MAXIMO System (EAM DISTRIBUTION)

Figure 4.4 is the result of As-Is and To Be analysis on the MAXIMO application system (EAM Distribution) where in the As Is condition the inspection function is owned by MJD (Banten Unit) and SIMANTRA (PPB Unit). Then the Monitoring function is available in the RENHAR application (North Sumatra Unit), P-MAN (East Kalimantan Unit), and JARVIST (East Kalimantan Unit). Furthermore, on the dashboard function, the RENHAR applications (North Sumatra Unit), AMADIS (West Sumatra Unit), SIMANTRA (North Sumatra Unit) are available.
PPB). The Asset Monitoring function is provided by RENHAR (North Sumatra Unit) and ASIIXMAX (East Java Unit 1). In the To Be condition, the system is centralized in the MAXIMO application (EAM Distribution). Where the inspection function already provides substation inspection features, substation master data, and load measurements along with their recapitulation. The Monitoring function provides maintenance planning features, maintenance monitoring, 20 KV network maintenance information, and network inspection. On the dashboard functions are unit leaderboards, individual leaderboards, planning, measuring substations, tree maps, reports, work programs, and backlogs. The monitoring function provides a health index feature and real-time inspection realization. The maintenance function provides features for overall maintenance effectiveness, overall maintenance for transformers, proactive maintenance for effective and breakdown maintenance. In the management function, there are vendor management and safety management features.

3.4 SCADA PORTAL Application Integration

The applications that will be integrated with the SCADA PORTAL application are SI-ODIS (North Sumatra Unit), BERSENADA (North Sumatra Unit), and MOOVES (West Sumatra Unit).

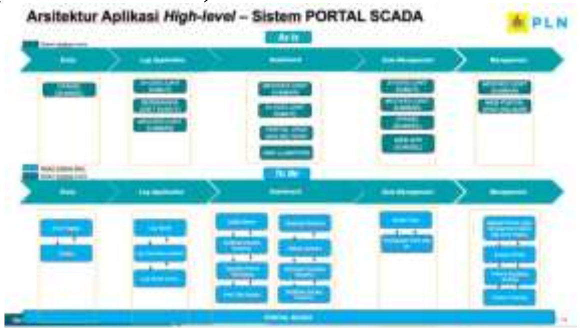

Figure 4.6 As Is and To Be Analysis of SCADA PORTAL Systems

Figure 4.6 is the result of As-Is and To Be analysis on the SCADA PORTAL application system where in As Is condition the entry function is provided by the CPANEL application (Sumsel Unit). The log application function is owned by SI-ODIS (Unit of North Sumatra), BERSENADA (Unit of North Sumatra), and MOOVES (Unit of West Sumatra). Furthermore, the dashboard function is available in the MOOVES application (West Sumatra Unit), the SI-ODIS application (North Sumatra Unit), UP2D PORTAL (Central Kalimantan Unit), and AMO (Lampung Unit). The data management function is provided by SI-ODIS (North Sumatra Unit), MOOVES (North Sumatra Unit), CPANEL (South Sumatra Unit), and WEB APD (South Sumatra Unit). The management function is provided by MOOVES (West Sumatra Unit) and UP2D WEB PORTAL (West Kalimantan Unit). In the To Be condition, the system is already centralized in the ARCGIS application. Where the log application function already provides scada logs and operating system logs. The dashboard function provides monthly voltage features, report recap, real-time equipment information, and real-time condition notifications. The monitoring function provides display features for feeder loads and a map of blackout points. The management function provides a mobile application for management when something goes out.

4. Conclusion

Enterprise Application Integration is suitable as an approach to integrate different applications into one integrated system within a company. System Level Design is a step that needs to be skipped in application development because it helps in designing the overall system architecture and includes planning from technical aspects, architecture and application design. Application architecture is used in application development because it is a fundamental that needs to be met is the design of an application or system that determines where the components are placed and how they communicate to support enterprise business functions.

Referensi

[1]` Renholm, K. (2011). Enterprise Application Integration: an Investigative

[2] Erasala, N., Yen, D. C., & Rajkumar, T. M. (2003). Enterprise application integration in the electronic commerce world. Computer Standards and Interfaces, 25(2), 69-82. https://doi.org/10.1016/S0925489(02)00106-X

[3] Irani, Z., Themistocleous, M., & Love, P. E.D. (2003). The impact of enterprise application integration on information system lifecycles Information 41(2),177-187Management, And https://doi.org/10.1016/S03787206(03)