

Design of Hospital Management Information System Pharmacy Module Using Total Architecture Synthesis Method (Study Case: University of Mataram Hospital)

M. Irfan Ramdhani, Dwi Ratnasari, Santi Ika Murpratiwi

Dept. Informatics Engineering, Mataram University

Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: mirfanr755@gmail.com, dwi.ratnasari@unram.ac.id, santiika@staff.unram.ac.id

**Corresponding Author*

Abstract This study aims to optimize the hospital pharmacy process at the University of Mataram Hospital by designing a user-friendly interface for the pharmacy module within the Hospital Management Information System (HMIS). In this research, a combined methodology of Total Architecture Synthesis (TAS) and User-Centered Design (UCD) was employed to ensure both a structured system architecture and a design that meets user needs. For the data collection observations, staff interviews, and literature reviews, which revealed several limitations and confusions in the existing system. The redesigned pharmacy module facilitates structured pharmacy workflow management and supports accurate data reporting. Usability evaluation using the User Experience Questionnaire (UEQ) yielded positive results across all dimensions, with "Stimulation" scoring highest at 1.68 on scale from -3 to +3. These findings indicate that the proposed system design is functional, engaging, and user-friendly, offering improved efficiency and usability in pharmaceutical service operations.

Key words: Hospital Management Information System, Total Architecture Synthesis, Pharmacy Module.

I. INTRODUCTION

The University of Mataram Hospital is a healthcare institution established in 2016 to serve the university's academic community and the surrounding population. As a developing teaching hospital, RS UNRAM continuously seeks to improve the quality and efficiency of its health services, including through the adoption of information systems to support operational and clinical processes.

The Pharmacy Installation is a service unit led by a pharmacist and serves as the place where all pharmaceutical activities are carried out. The core tasks of pharmacy management essentially include planning, procurement, storage, dispensing of medication based on prescriptions for inpatient and outpatient care, quality control, and distribution. This pharmacy installation also ensures that all pharmaceutical processes comply with government regulations and hospital accreditation standards [1].

The Hospital Information Management System is a computer-based system that processes and connects all business processes in healthcare services through coordination, reporting, and administrative procedures. The goal is to obtain information quickly, accurately, and precisely [2]. Currently, computerized Management Information Systems in hospitals have become essential tools to facilitate the operational management of hospitals. The success of system development is an investment to improve patient service quality. To meet these service demands, support from a management information system is essential.

Based on the interview conducted, the Pharmacy Installation at the University of Mataram Hospital has already used HMIS Khanza, a hospital management information system. This system already has features integrated with other service modules and is even integrated with BPJS Application. However, the system's interface is considered complicated, which sometimes makes it difficult to use. The merging of pharmacist and warehouse staff functionalities into a single interface reflects not only a poor user interface design, but also an inadequate role-based workflow separation. This conflation leads to operational confusion, inefficient task execution, and an increased risk of data input errors, all of which impact the overall performance and reliability of pharmacy services. Therefore, addressing these challenges requires more than a superficial redesign of the user interface.

One relevant study was conducted by Rahayu et al. (2024), who performed a heuristic evaluation of a pharmacy surveillance information system using Nielsen's usability principles. The study identified 91 usability issues, primarily related to consistency, layout clarity, and error prevention. The most critical problems involved overly dense interfaces, poor intuitiveness, and lack of user feedback. These findings suggest that pharmacy information systems not designed with usability in mind can hinder user performance and increase the likelihood of input errors.

This study proposes a design of a hospital management information system on pharmacy module, specifically focusing on role-based interface segregation, where pharmacist and warehouse staff functionalities are clearly separated into distinct user interfaces tailored to their specific workflows and responsibilities. This approach will eliminate interface confusion, reduce navigation complexity, and minimize input errors by providing each user role with streamlined, task-focused interfaces. Furthermore, the system will maintain seamless integration capabilities with other hospital modules such as payment systems, inpatient and outpatient services, scheduling, registration, and electronic medical records, while ensuring compatibility with external systems like BPJS Health.

To ensure that the redesigned pharmacy module meets both organizational goals and user expectations, this study adopts a combined methodology of Total Architecture Synthesis (TAS) and User-Centered Design (UCD). TAS is utilized to guide the overall architectural design process through structured phases. This method ensures that the system is developed based on a comprehensive understanding of organizational workflows and technical requirements, which is essential in a healthcare environment that demands accuracy, integration, and compliance. Complementing TAS, the UCD approach is employed to prioritize the experience and needs of end-users—namely pharmacists and warehouse staff—through active involvement during design, prototyping, and usability testing. UCD ensures that the resulting user interface is intuitive, role-specific, and aligned with real-world pharmacy operations. The integration of TAS and UCD allows the system to be robust at the architectural level while remaining user-friendly and operationally practical. Furthermore, the system design also adopts all five principles of the SOLID object-oriented design paradigm such as Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, and Dependency Inversion to enhance the modularity, flexibility, and scalability of the system architecture. These principles guide the structure of components, classes, and service interactions, ensuring that each module has a clear purpose, is easily extendable without modifying existing code, and promotes loose coupling through the use of abstraction. By implementing SOLID principles alongside TAS and UCD, the resulting system design is not only technically robust and user-centered but also maintainable and adaptable to future development needs.

To evaluate the usability and overall user experience of the redesigned pharmacy module interface, this study employs the User Experience Questionnaire (UEQ) as the primary testing method. UEQ is specifically designed to measure user perceptions across multiple dimensions of user experience, including attractiveness, clarity, efficiency, dependability, stimulation, and novelty. This makes it highly suitable for assessing interactive systems in complex environments such as healthcare. UEQ

provides a comprehensive view by capturing users' emotional responses and satisfaction levels during system interaction. Moreover, UEQ supports rapid quantitative analysis and benchmarking against other systems, enabling researchers to identify specific areas of strength and weakness in the interface design. Given the goal of developing a user-centered and intuitive system, the use of UEQ ensures that the final prototype not only functions correctly but also deliver a positive and engaging experience for users. This design is expected to produce a system that is not only efficient and functional, but also easy to use, thereby simplifying the pharmacy installation management process at the University of Mataram Hospital.

II. LITERATURE REVIEW

The design of hospital management information system pharmacy module is based on previous research, which serve as the main reference in designing the architecture system which integrate with other modules in hospital management information system as follows:

In a study titled "Designing a Hospital Management Information System Pharmacy Module," the system was developed using Total Architecture Synthesis (TAS) to generate designs that align with the structured programming concept. The research aims to improve data processing, inventory management, drug compounding, pricing, sales, and reporting in hospital pharmacies. The system integrates various processes and modules, ensuring faster, more accurate information for decision-making. The design includes context diagrams, data flow diagrams, and user interface designs, which are evaluated for consistency with user requirements [1].

In a study titled "Designing a Hospital Management Information System Service Module," the design process used Total Architecture Synthesis (TAS) to create an integrated system for managing various hospital processes, including master data management, treatment, emergency services, medical records, scheduling, and reporting. The system design aimed to reduce manual processes and improve data exchange between modules to enhance decision-making efficiency. The result of this design includes key elements such as context diagrams, data flow diagrams, hierarchy charts, physical data models, and graphical user interfaces. The application of TAS in this study demonstrates its effectiveness in creating a well-structured and integrated hospital information system [5].

In a study titled "Implementation of UML Diagrams in Designing Inpatient Patient Data Application at Puskesmas Lubuk Buaya," Unified Modeling Language (UML) was used to design the system, which helped in developing a more organized and efficient patient data management process. The research aimed to create a system that would replace the manual data entry process, which had been prone to errors and inefficiencies. The result was a fully functional application that streamlined patient data management, reduced redundancy, and provided faster, more reliable reports, ultimately

enhancing the decision-making process for healthcare providers [6].

In a study titled "UI/UX Design for Bank Jago Application Using User-Centered Design Method," the researcher applied User-Centered Design (UCD) to create a user-friendly interface for an insurance feature on the Bank Jago mobile application. The study focused on ensuring usability through principles like learnability, efficiency, and satisfaction. By involving users in every stage of the design process, the approach helped develop a prototype that was well-received in terms of usability. Applying UCD in the design of SIMRS pharmacy modules can similarly enhance the user interface, making it easier for pharmacy staff to manage data and improve the overall efficiency of the system, ensuring better acceptance and usability in hospital settings [4].

In Regulation of the Minister of Health Number 58 of 2014 on Pharmaceutical Services Standards in Hospitals, the definition of a hospital is provided as a health service institution that delivers comprehensive individual healthcare services, including inpatient, outpatient, and emergency services. The regulation emphasizes the importance of hospital infrastructure, human resources pharmaceutical services, and equipment to ensure quality healthcare. Specifically, Pharmacy Installation is defined as a functional unit responsible for managing all pharmaceutical services within a hospital. This includes the management of pharmaceutical supplies, medical equipment, and consumables, ensuring quality, safety, and efficacy in their use. The regulation underscores that pharmacy services in hospitals must adhere to specific standards to guarantee patient safety and improve healthcare outcomes. Therefore, in the design of a Hospital Information Management System (SIMRS), it is essential to integrate the functions of Pharmacy Installation with other hospital modules to optimize operational efficiency and meet standards for pharmaceutical service [7].

III. METHODOLOGY

This study adopts an integrated methodology combining Total Architecture Synthesis (TAS) and User-Centered Design (UCD) to design a hospital pharmacy module that is both structurally sound and user-oriented. TAS is applied in the early stages to define system scope, identify technical and business requirements, model business processes, and design a scalable and modular system architecture. Simultaneously, UCD is incorporated within and after each TAS stage to ensure that user needs and usability principles are integrated throughout the design.

Specifically, during the *requirement analysis* phase of TAS, UCD techniques such as direct observation and user interviews were conducted to gather insights from pharmacists and warehouse staff. During the business process modeling and system architecture design, user feedback was used to validate workflows, task sequences, and feature prioritization. In the UI/UX design stage, high-fidelity prototypes were built and iteratively tested with

users using UCD usability testing techniques. Thus, while TAS ensures a structured, technically feasible system design, UCD ensures the system is usable and relevant to real-world user tasks.

This cohesive methodological framework ensures that the resulting pharmacy module is well-structured, user-friendly, and aligned with both institutional requirements and real-world usage.

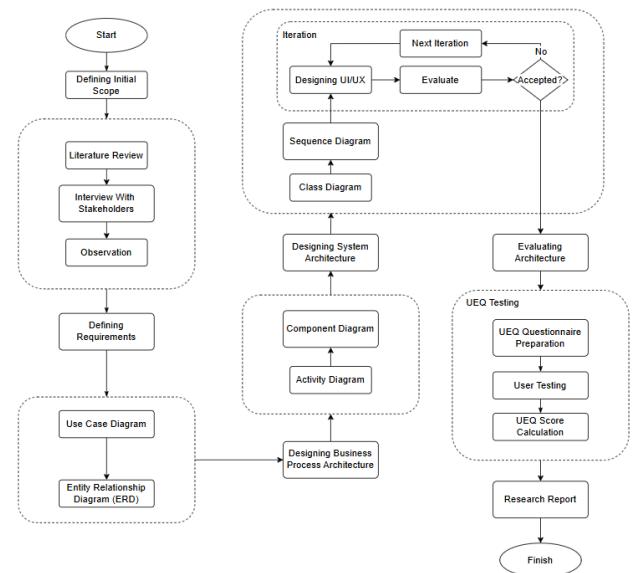


Fig. 1. Research Flow

A. Defining initial scope

This stage aims to define the boundaries and objectives of the software development project in broad terms. It involves an initial understanding of the problem to be solved, the underlying business needs, and the expected outcomes from the software system to be developed.

B. Defining requirements

This stage involves the collection and detailed, structured analysis of user and business requirements. The goal is to gain a comprehensive understanding of what users expect from the software system and identify both functional and non-functional requirements that must be fulfilled.

C. Designing business process architecture

This stage focuses on modeling and analyzing the business processes of the organization that will be supported or influenced by the developed software. The goal is to understand the workflow, interdepartmental interactions, and how information is processed within the organization.

D. Designing system architecture

This stage is the core of the software design process, where the system's technical architecture is defined. It includes the selection of technologies, database design, user interface (UI) and user experience (UX) design.

Additionally, the pharmacy module is designed to integrate with other modules using RESTful APIs and JSON data formats. A microservices architecture is adopted to ensure seamless data exchange between modules such as medical records and billing, enabling flexibility, scalability, and easier maintenance. integration with other systems.

In this study, the designed system architecture results in Entity Relationship Diagram (ERD), Activity Diagram, Component Diagram, Class Diagram, Sequence Diagram, and UI/UX design. Furthermore, the system design also adopts all five principles of the SOLID object-oriented design to enhance the modularity, flexibility, and scalability of the system architecture. These principles guide the structure of components, classes, and service interactions, ensuring that each module has a clear purpose. To approach the results, UCD plays a vital role in ensuring that the design of the UI/UX is truly aligned with the needs, preferences, and working context of end-users. In the UCD approach, users are actively involved from the early stages through observation, interviews, prototype testing, and iterative feedback, so that every design element is developed based on a real understanding of user behavior and challenges. The application of UCD allows the resulting system to be not only technically functional but also intuitive, easy to use, and supportive of daily operational efficiency.

E. Evaluating architecture

To evaluate the quality of the user interface and overall user experience of the system prototype, the *User Experience Questionnaire* (UEQ) is utilized. This method aims to assess users' subjective impressions and satisfaction when interacting with the system interface. UEQ consists of 8 or 26 items divided into six dimensions: Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation, and Novelty. Each item is presented as a pair of contrasting adjectives (e.g., "complicated – easy", "boring – exciting") and is rated by respondents using a 7-point Likert scale[5]. Although UEQ is typically used on operational systems, in this study it was adapted to evaluate a high-fidelity prototype using scenario-based simulations. Users were instructed to complete specific tasks such as stock updates and prescription validation through a clickable prototype. The evaluation was conducted immediately after the task completion to capture their impressions, making the UEQ suitable for usability testing. This research uses the 26 items of UEQ that can be seen in Figure 2.

	1	2	3	4	5	6	7		
menyusahkan	○	○	○	○	○	○	○	menyenangkan	1
tak dapat dipahami	○	○	○	○	○	○	○	dapat dipahami	2
kreatif	○	○	○	○	○	○	○	monoton	3
mudah dipelajari	○	○	○	○	○	○	○	sulit dipelajari	4
bermanfaat	○	○	○	○	○	○	○	kurang bermanfaat	5
membosankan	○	○	○	○	○	○	○	mengasyikkan	6
tidak menarik	○	○	○	○	○	○	○	menarik	7
tak dapat diprediksi	○	○	○	○	○	○	○	dapat diprediksi	8
cepat	○	○	○	○	○	○	○	lambat	9
berdaya cipta	○	○	○	○	○	○	○	konvensional	10
menghalangi	○	○	○	○	○	○	○	mendukung	11
baik	○	○	○	○	○	○	○	buruk	12
rumit	○	○	○	○	○	○	○	sederhana	13
tidak disukai	○	○	○	○	○	○	○	menggembirakan	14
lazim	○	○	○	○	○	○	○	terdepan	15
tidak nyaman	○	○	○	○	○	○	○	nyaman	16
aman	○	○	○	○	○	○	○	tidak aman	17
memotivasi	○	○	○	○	○	○	○	tidak memotivasi	18
memenuhi ekspektasi	○	○	○	○	○	○	○	tidak memenuhi ekspektasi	19
tidak efisien	○	○	○	○	○	○	○	efisien	20
jelas	○	○	○	○	○	○	○	membungkan	21
tidak praktis	○	○	○	○	○	○	○	praktis	22
terorganisasi	○	○	○	○	○	○	○	berantakan	23
atraktif	○	○	○	○	○	○	○	tidak atraktif	24
ramah pengguna	○	○	○	○	○	○	○	tidak ramah pengguna	25
konservatif	○	○	○	○	○	○	○	inovatif	26

Fig. 2. UEQ Questions

The reliability testing of the questionnaire data was conducted using the Cronbach's alpha coefficient. This coefficient reflects the consistency across all UX factor scales. The test results show that the Cronbach's alpha values for all UX factors exceed the acceptable threshold (>0.7), namely: attractiveness (0.9), perspicuity (0.79), efficiency (0.88), dependability (0.83), stimulation (0.81), and novelty (0.83). These results indicate that the data analyzed using the UEQ method are reliable.

UEQ data can be processed using the UEQ analysis tool. To interpret the values obtained from each UEQ scale, the calculated scores are compared with the values in Table 1 below.

TABLE 1. Benchmark Interval for UEQ

	Excellent	Good	Above Average	Below Average	Bad
Attractiveness	$\geq 1,75$	$\geq 1,52$ $< 1,75$	$\geq 1,17$ $< 1,52$	$\geq 0,7$ $< 1,17$	$< 0,7$
Perspicuity	$\geq 1,9$	$\geq 1,56$ $< 1,9$	$\geq 1,08$ $< 1,56$	$\geq 0,64$ $< 1,08$	$< 0,64$
Efficiency	$\geq 1,78$	$\geq 1,47$ $< 1,78$	$\geq 0,98$ $< 1,47$	$\geq 0,54$ $< 0,98$	$< 0,54$
Dependability	$\geq 1,65$	$\geq 1,48$ $< 1,65$	$\geq 1,14$ $< 1,48$	$\geq 0,78$ $< 1,14$	$< 0,78$
Stimulation	$\geq 1,55$	$\geq 1,31$ $< 1,55$	$\geq 0,99$ $< 1,31$	$\geq 0,5$ $< 0,99$	$< 0,5$
Bad	$\geq 1,4$	$\geq 1,05$ $< 1,4$	$\geq 0,71$ $< 1,05$	$\geq 0,3$ $< 0,71$	$< 0,3$

The analysis results will help identify areas of strength and weakness in the user interface design, and will serve as a basis for refinement in subsequent development stages. By applying the UEQ method, the design process remains user-centered, ensuring that the final system aligns with user expectations in terms of usability and experience.

In this study, UEQ was adapted to evaluate a high-fidelity prototype using scenario-based simulations. Users were instructed to complete specific tasks such as stock updates and prescription validation through a clickable

B.1. Use Case Diagram

In the hospital management information system, specifically in the pharmacy module, there are several key actors involved in the system's operational flow: the Pharmacist, Warehouse Staff, Pharmacy Assistant, and the Head of the Pharmacy Department. Each actor has distinct roles and responsibilities designed to ensure the security, structure, and efficiency of pharmaceutical processes.

This use case diagram comprehensively illustrates the relationships and interactions between the system and its actors, outlining the main operational flows such as inventory management, pharmaceutical transactions, procurement and distribution, and master data management, all of which support the core functions of the hospital pharmacy module. This structured division of roles allows the system to maintain data integrity, ensure accountability, and support streamlined workflow in pharmaceutical services, while also aligning with hospital regulations and accreditation standards. The activities of each user, as outlined in the Use Case Diagram, are shown in Figure 3.

The TAS method begins with determining the initial scope by defining the business scope. Based on interviews with the stakeholders of pharmacy installation at University of Mataram including pharmacy staff and the warehouse staff, the general scope of work in the Pharmacy Installation Unit at the University of Mataram Hospital includes providing medication services to patients and managing the inventory of medical supplies. The general scope of the designed pharmacy module in the Hospital Management Information System (HMIS) of Mataram University Mataram is as follows.

Main Scope	Description
Master Data Management	Manages core data such as lists of medicines, medical equipment, consumables, suppliers, units, and item categories.
Medical Inventory Management	Covers recording of incoming and outgoing items, stock monitoring, reorder point settings, and processes such as stock-taking and expiration tracking.
Patient Medical Item Services	Supports the process of requesting, verifying, and dispensing medications/medical items to patients based on prescriptions or medical instructions.

```

    usecaseDiagram
        actor Pharmacist
        actor Head of Pharmacy
        actor Pharmacist Assistant

        usecase U1[Electronic Medical Record (EMR) Module]
        usecase U2[Prescription Verification]
        usecase U3[Prescription Update]
        usecase U4[Print Prescription]
        usecase U5[Prescription Compounding]
        usecase U6[Prescription Planning]
        usecase U7[Input Purchasing Invoice]
        usecase U8[Purchase Order]
        usecase U9[Purchase Planning History]
        usecase U10[Medicines and Medical Equipment Procurement]
        usecase U11[Purchase Order History]
        usecase U12[Login]
        usecase U13[CRUD medicines and medical equipment data]
        usecase U14[CRUD supplier data]
        usecase U15[Master Data Management]
        usecase U16[Medical Equipment Purchase Approval]
        usecase U17[Medical Dispensing]
        usecase U18[Prescription Retrieval]
        usecase U19[Payment Module]
        usecase U20[Inventory Taking]
        usecase U21[Warehouse Staff]
        usecase U22[Dispensing History]
        usecase U23[Inpatient and Outpatient Care Module]
        usecase U24[logout]

        Pharmacist --> U5
        Pharmacist --> U4
        Pharmacist --> U2
        Pharmacist --> U3
        Pharmacist --> U1
        Pharmacist --> U6
        Pharmacist --> U7
        Pharmacist --> U8
        Pharmacist --> U9
        Pharmacist --> U10
        Pharmacist --> U11
        Pharmacist --> U12
        Pharmacist --> U13
        Pharmacist --> U14
        Pharmacist --> U15
        Pharmacist --> U16
        Pharmacist --> U17
        Pharmacist --> U18
        Pharmacist --> U19
        Pharmacist --> U20
        Pharmacist --> U21
        Pharmacist --> U22
        Pharmacist --> U23
        Pharmacist --> U24

        Head of Pharmacy --> U15
        Head of Pharmacy --> U16
        Head of Pharmacy --> U17
        Head of Pharmacy --> U18
        Head of Pharmacy --> U19
        Head of Pharmacy --> U20
        Head of Pharmacy --> U21
        Head of Pharmacy --> U22
        Head of Pharmacy --> U23
        Head of Pharmacy --> U24

        Pharmacist Assistant --> U17
        Pharmacist Assistant --> U18
        Pharmacist Assistant --> U19
        Pharmacist Assistant --> U20
        Pharmacist Assistant --> U21
        Pharmacist Assistant --> U22
        Pharmacist Assistant --> U23
        Pharmacist Assistant --> U24

        U5 --> U4 : <<extend>>
        U5 --> U2 : <<extend>>
        U5 --> U3 : <<extend>>
        U5 --> U1 : <<include>>
        U5 --> U6 : <<include>>
        U5 --> U7 : <<include>>
        U5 --> U8 : <<include>>
        U5 --> U9 : <<include>>
        U5 --> U10 : <<include>>
        U5 --> U11 : <<include>>
        U5 --> U12 : <<include>>
        U5 --> U13 : <<include>>
        U5 --> U14 : <<include>>
        U5 --> U15 : <<include>>
        U5 --> U16 : <<include>>
        U5 --> U17 : <<include>>
        U5 --> U18 : <<include>>
        U5 --> U19 : <<include>>
        U5 --> U20 : <<include>>
        U5 --> U21 : <<include>>
        U5 --> U22 : <<include>>
        U5 --> U23 : <<include>>
        U5 --> U24 : <<include>>

        U12 --> U13 : <<include>>
        U12 --> U14 : <<include>>
        U12 --> U15 : <<include>>
        U12 --> U16 : <<include>>
        U12 --> U17 : <<include>>
        U12 --> U18 : <<include>>
        U12 --> U19 : <<include>>
        U12 --> U20 : <<include>>
        U12 --> U21 : <<include>>
        U12 --> U22 : <<include>>
        U12 --> U23 : <<include>>
        U12 --> U24 : <<include>>

        U13 --> U14 : <<include>>
        U13 --> U15 : <<include>>
        U13 --> U16 : <<include>>
        U13 --> U17 : <<include>>
        U13 --> U18 : <<include>>
        U13 --> U19 : <<include>>
        U13 --> U20 : <<include>>
        U13 --> U21 : <<include>>
        U13 --> U22 : <<include>>
        U13 --> U23 : <<include>>
        U13 --> U24 : <<include>>

        U14 --> U15 : <<include>>
        U14 --> U16 : <<include>>
        U14 --> U17 : <<include>>
        U14 --> U18 : <<include>>
        U14 --> U19 : <<include>>
        U14 --> U20 : <<include>>
        U14 --> U21 : <<include>>
        U14 --> U22 : <<include>>
        U14 --> U23 : <<include>>
        U14 --> U24 : <<include>>

        U15 --> U16 : <<include>>
        U15 --> U17 : <<include>>
        U15 --> U18 : <<include>>
        U15 --> U19 : <<include>>
        U15 --> U20 : <<include>>
        U15 --> U21 : <<include>>
        U15 --> U22 : <<include>>
        U15 --> U23 : <<include>>
        U15 --> U24 : <<include>>

        U16 --> U17 : <<include>>
        U16 --> U18 : <<include>>
        U16 --> U19 : <<include>>
        U16 --> U20 : <<include>>
        U16 --> U21 : <<include>>
        U16 --> U22 : <<include>>
        U16 --> U23 : <<include>>
        U16 --> U24 : <<include>>

        U17 --> U18 : <<include>>
        U17 --> U19 : <<include>>
        U17 --> U20 : <<include>>
        U17 --> U21 : <<include>>
        U17 --> U22 : <<include>>
        U17 --> U23 : <<include>>
        U17 --> U24 : <<include>>

        U18 --> U19 : <<include>>
        U18 --> U20 : <<include>>
        U18 --> U21 : <<include>>
        U18 --> U22 : <<include>>
        U18 --> U23 : <<include>>
        U18 --> U24 : <<include>>

        U19 --> U20 : <<include>>
        U19 --> U21 : <<include>>
        U19 --> U22 : <<include>>
        U19 --> U23 : <<include>>
        U19 --> U24 : <<include>>

        U20 --> U21 : <<include>>
        U20 --> U22 : <<include>>
        U20 --> U23 : <<include>>
        U20 --> U24 : <<include>>

        U21 --> U22 : <<include>>
        U21 --> U23 : <<include>>
        U21 --> U24 : <<include>>

        U22 --> U23 : <<include>>
        U22 --> U24 : <<include>>

        U23 --> U24 : <<include>>
    
```

The use case diagram illustrates the key functionalities of the Pharmacy Module within the Hospital Management Information System (HMIS) at University of Mataram Hospital. It involves four main actors: Pharmacist, Warehouse Staff, Pharmacy Head, and Pharmacy Assistant, each interacting with the system based on their roles. The Pharmacist is primarily responsible for the dispensing of medications, which includes extended functions such as confirming prescriptions, updating prescription status, preparing medications, and printing prescriptions. Pharmacists can also view current drug stock to ensure availability. The Warehouse Staff is in charge of inventory-related operations such as procurement and inventory checking. Procurement includes tasks like creating purchase plans, tracking orders, and entering purchase invoices. The Pharmacy Head has authority over data governance and decision approval processes. This includes managing all master data, such as CRUD

At this stage, the system requirements were specified based on the discussions held with the pharmacist and the pharmacy installation staff at University of Mataram Hospital. The identified requirements are the end-user of

operations for drugs, suppliers, and medical equipment/consumables, and approving drug purchases and inventory mutations. Lastly, the Pharmacy Assistant supports the medication dispensing process through the prescription filling use case and medical dispensing. All actors access the system through common functions like login and logout, ensuring secure and role-based access to the system features.

B.2. Entity Relationship Diagram

Based on the use case diagram presented in Figure 3, an overview of each user's functions within the system has been obtained. From this diagram, the required database for the hospital management information system, specifically the pharmacy module, can be determined. The database design utilized in this study is represented by an Entity Relationship Diagram (ERD). The ERD illustrated in Figure 4 and Figure 5.

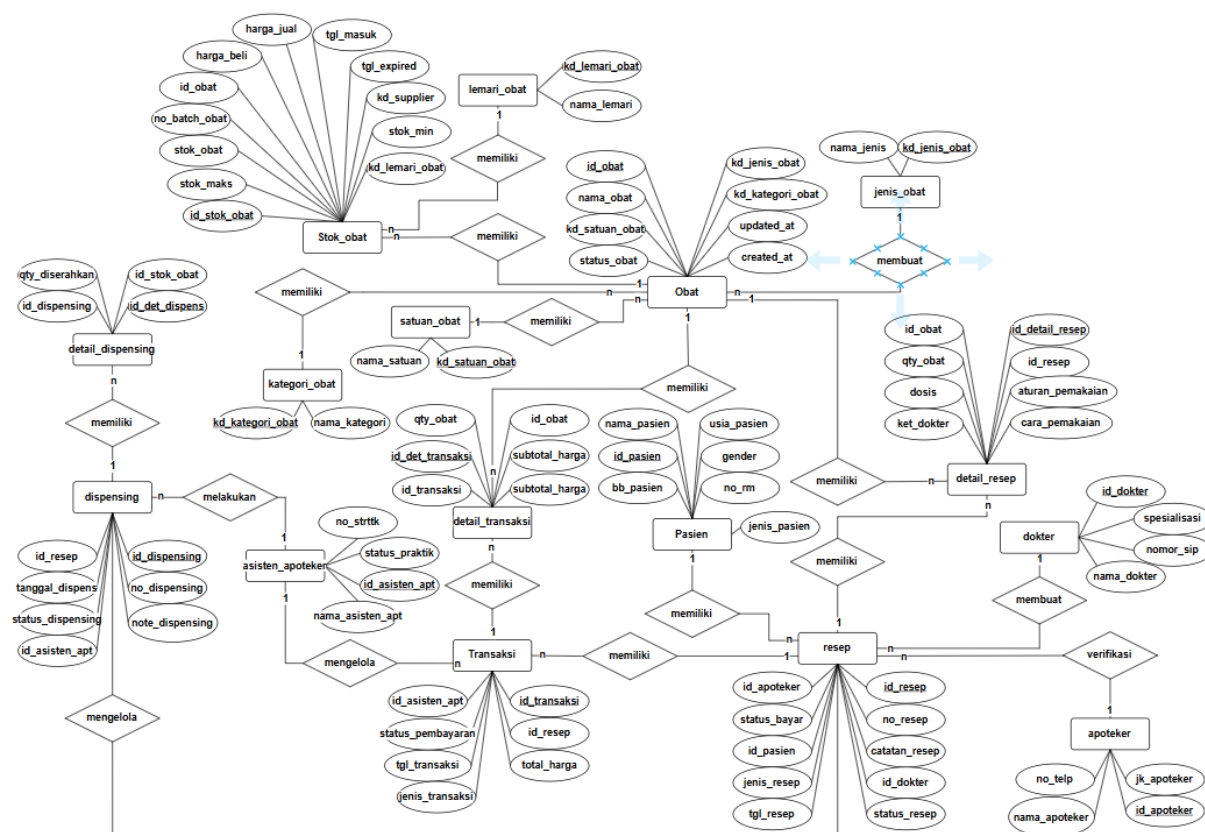


Fig. 4. Entity Relationship Diagram (ERD) – Part 1

The ERD shown in Figure 4 illustrates the integrated data relationship structure within the pharmacy module of the Hospital Information System (HIS) for patient medication services. The system consists of master data entities including the drug entity that stores fundamental information such as drug names, types, and classifications, as well as the drug stock entity that manages operational data including stock quantities, expiration dates, batch numbers, and prices. The separation of these two entities

aims to optimize database normalization, enabling the management of multiple drug batches with different expiration dates. The system is equipped with operational entities to record clinical activities including prescriptions from physicians, drug dispensing by pharmacists, and drug distribution transactions to patients. The integration of all these entities generates a complete audit trail from prescription writing to drug delivery, supporting patient safety and compliance with pharmaceutical service standards.

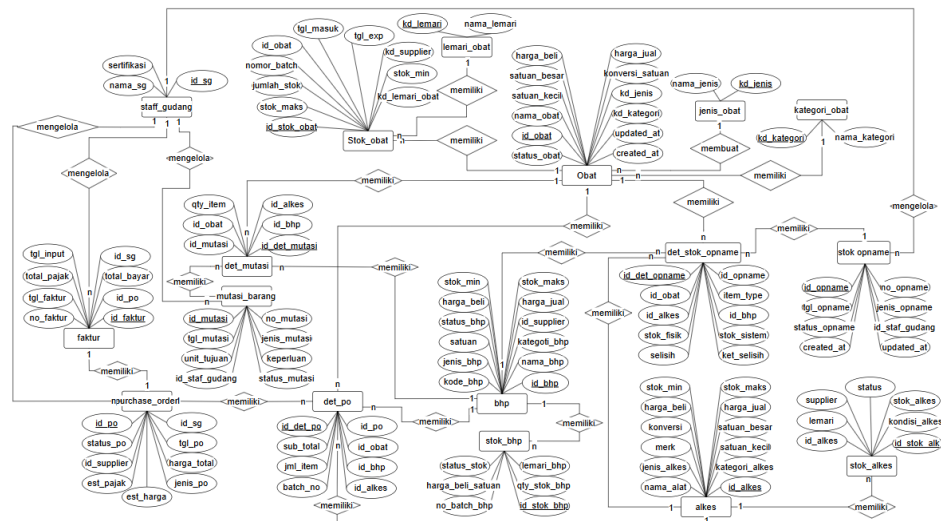


Fig. 5. Entity Relationship Diagram (ERD) – Part 2

The ERD in Figure 5 illustrates a comprehensive structure of data relationship flows for medical goods management systems that encompasses three main categories: pharmaceuticals, medical devices (alkes), and consumable medical supplies (BHP) within the hospital environment. This system is designed to integrate all aspects of medical supply chain management from procurement to distribution, considering the unique characteristics of each type of medical goods that have different regulations and storage standards. This data model implements three main entities to systematically and structurally record inventory management activities. The stock opname entity functions as an audit and physical inventory verification mechanism conducted periodically to ensure the accuracy of stock data in the system with real conditions in the field, including recording stock discrepancies, causes of discrepancy, and necessary corrective actions. The medical goods procurement entity is designed to manage the entire procurement process from needs planning, purchase order issuance, and goods receipt.

C. Designing business process architecture

At this stage, business process and system architecture design are carried out in the form of Activity Diagram and Component Diagram. Activity Diagram provides a detailed illustration of the business processes occurring at the Pharmacy Installation of the University of Mataram Hospital. The component diagram provides an overview of the integration flow between modules in the UNRAM Hospital Information System (SIMRS).

C.1. Activity Diagram

The Activity Diagram is a detailed representation of the activities identified in the Use Case Diagram. Below is an example of one of the detailed activities from the Use Case Diagram, presented in the form of an Activity Diagram.

Warehouse Staff

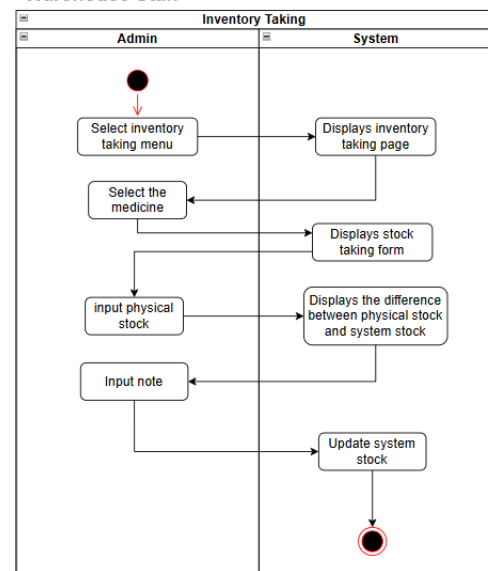


Fig. 6. Activity Diagram

This activity diagram shown in Figure 6 illustrates the stock opname process conducted by warehouse staff in the hospital inventory management system. The process begins with initiation by the admin, followed by preparation and physical counting of goods by warehouse staff. After counting is completed, the system performs validation by comparing the physical count results against system data. If discrepancies exist, the system generates a discrepancy report and warehouse staff conduct an investigation into the causes of the discrepancy, then update the system data according to actual conditions. If there are no discrepancies, the process directly proceeds to stock rotation updates. This entire workflow is designed to ensure inventory data accuracy and support effective internal control in warehouse management, with decision points that enable systematic handling of stock differences.

C.2. Component Diagram

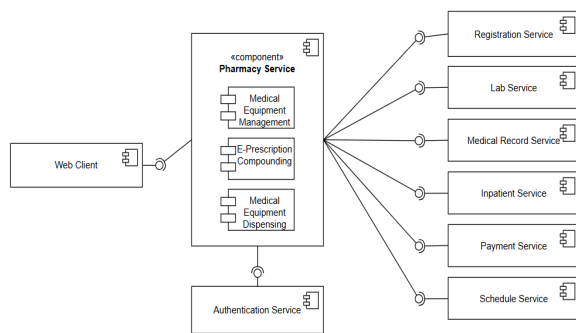


Fig. 7. Component Diagram

The diagram shown in Figure 7 illustrates a microservices-based architecture for a Hospital Management Information System (SIMRS), where each service is separated into distinct components to enhance system scalability and flexibility. The main focus is on the *Pharmacy Service*, which consists of three sub-components: *Medical Equipment Management*, *Prescription Compounding*, and *Medical Equipment Dispensing*, each handling specific functions within the pharmacy module. The Pharmacy Service interacts with various other services such as *Registration Service*, *Lab Service*, *Medical Record Service*, *Inpatient Service*, *Payment Service*, and *Schedule Service* to support a comprehensive pharmacy workflow. All communication with the system is conducted through the *Web Client*, while user authentication is handled by the *Authentication Service*. This microservices approach allows each service to operate independently and communicate via RESTful APIs, making the system easier to manage, develop, and maintain in a centralized yet distributed manner.

D. Designing System Architecture

At this stage, the system architecture design is carried out based on the business process architecture that was previously created. The system architecture formed in this stage consists of Class Diagrams, Sequence Diagrams, and UI/UX design. The SOLID principles is implemented in this part of the research. Single Responsibility Principle (SRP) is applied by ensuring that each class in the Class Diagram is responsible for only one functional area (e.g., managing stock, verifying prescriptions). Open/Closed Principle (OCP) is reflected in the modular design, allowing new features (such as additional validation layers or inventory rules) to be added without altering existing components. Liskov Substitution Principle (LSP) is applied through inheritance hierarchies that ensure subclass objects can replace parent classes without disrupting system behavior. Interface Segregation Principle (ISP) is applied by creating specific interfaces for different user roles (pharmacists, warehouse staff, pharmacy assistants), as seen in both the Use Case and UI/UX design. Dependency Inversion Principle (DIP) is implemented by decoupling high-level modules (like UI) from low-level modules (like database access) using interface-based service interactions and abstraction layers in the microservices architecture.

D.1. Class Diagram

Based on the Entity Relationship Diagram (ERD) and Activity Diagram that have been designed, a Class Diagram is then developed to complete the system design. The ERD serves as the foundation for identifying the entities, their attributes, and the relationships between those entities, all of which are translated into classes within the Class Diagram. The detailed activities depicted in the Activity Diagram will be translated into methods for each class that has been identified in the Class Diagram. These methods define the behavior of the objects created from the classes and outline how they interact with the system's data and processes [9]. The Class Diagram developed in this study can be seen in Figure 7.

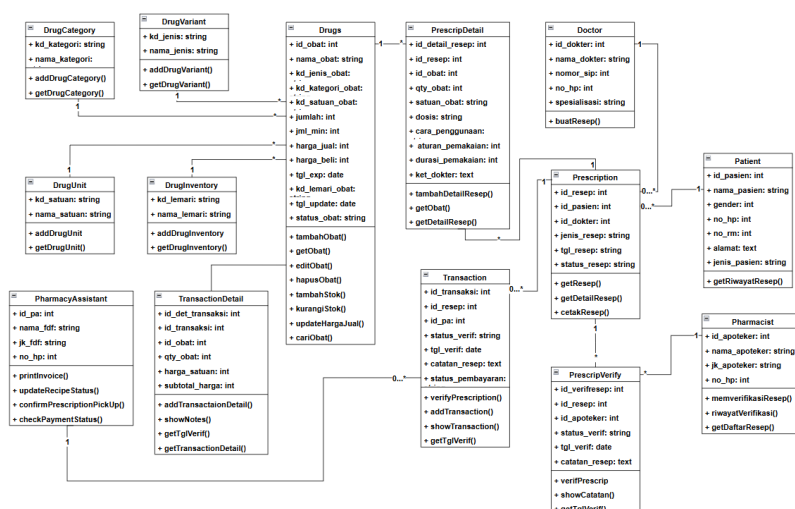


Fig.7. Class Diagram – Part 1

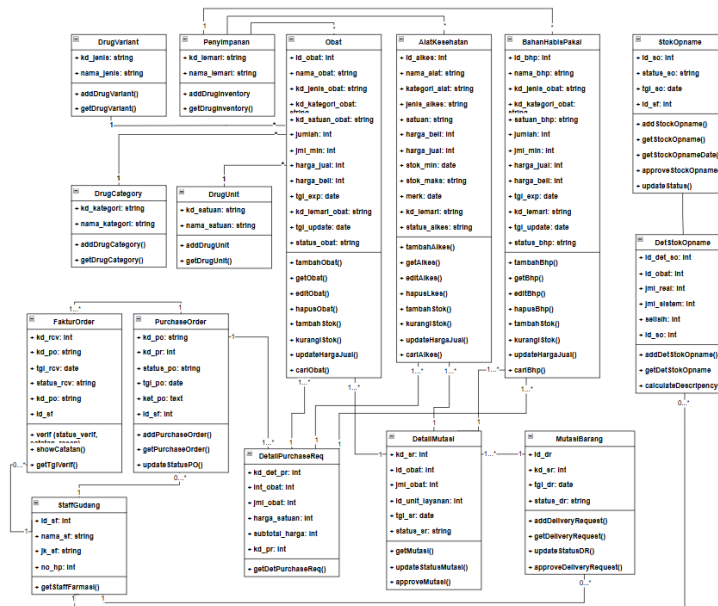


Fig.8. Class Diagram – Part 2

D.2. Sequence Diagram

Based on the previously created Class Diagram, a blueprint has been obtained that can be used to build the objects in the system. Then, to illustrate how these objects interact with each other in a more detailed system flow, a Sequence Diagram is needed. The Sequence Diagram visualizes the order of interactions between objects within the system, showing how objects communicate and exchange messages over time, in accordance with the flow depicted in the Activity Diagram [9]. One example of the Sequence Diagram used in this study can be seen in Figure 8, which illustrates the interaction between objects in one of the system's processes.

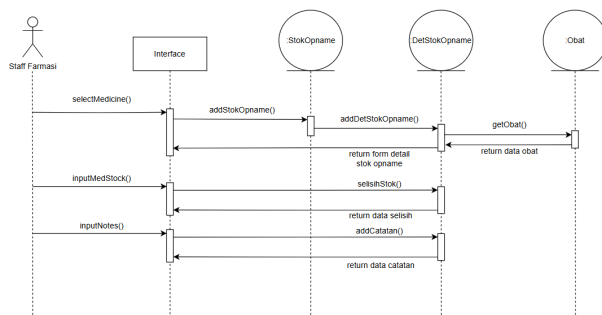


Fig. 8. Sequence Diagram

This sequence diagram shown in Figure 8 illustrates the stock opname process involving Staff Farmasi, Interface, StokOpname controller, DatabaseOpname, and Obat entity. The process begins when Staff Farmasi accesses stock opname data through the interface, which then retrieves information from the database and drug entity to be displayed to the user. Next, Warehouse Staff inputs the physical count results through the inputMedStock() function. The StokOpname controller processes and

validates this data by comparing the physical count results against system data. After successful validation, the system saves the results to the database and provides confirmation to Staff Farmasi. This sequence ensures data integrity through a validation layer that verifies consistency between manual counting and system data to support accurate pharmacy inventory management.

D.3. UI/UX Design

The final stage in this system design is the UI/UX (User Interface/User Experience) design. This phase focuses on creating an interface that is not only functional but also intuitive and user-friendly. The design process actively involves end users to gather feedback and insights, ensuring that the interface meets their expectations and aligns with their daily workflows. By adopting a User-Centered Design (UCD) approach, user involvement is integrated at each step so that the resulting interface is both efficient and easy to navigate. Below is an example of a UI design tailored for one of the user roles within the Pharmacy Module, demonstrating how interface elements are adapted to support the specific tasks and responsibilities of that user.

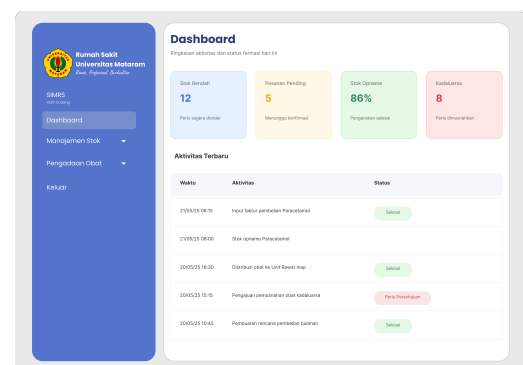


Fig. 9. Dashboard Page of Warehouse Staff

The dashboard for Warehouse Staff is specifically designed to display a summary of important notifications related to drug inventory and medication procurement. This includes alerts for low stock levels and information on medications that are nearing or have passed their expiration date. By providing this summarized view, Warehouse Staff no longer need to manually check each individual drug item, thereby saving time and reducing the risk of overlooking critical stock issues. This type of interface design aims to support efficiency and responsiveness in daily operations. With real-time alerts and quick access to essential information, staff can immediately take appropriate actions.

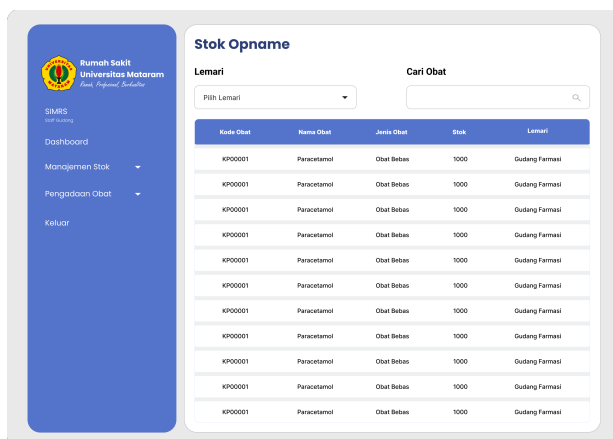


Fig. 10. Inventory Taking Page of Warehouse Staff

The UI shown in Figure 10 shows the user interface (UI) of the Inventory Taking page in the pharmacy stock management module of the Hospital Management Information System (SIMRS) at Mataram University. The main section of the page displays a list of medicine stock data focused on the stock opname process. There are options to select a storage cabinet and search for medicines by name. The table below provides detailed information about the medicine stock, including the drug code (e.g., KPO0001), drug name (Paracetamol), drug type (Over-the-Counter), stock quantity (1000), and storage location (Pharmacy Warehouse). When an item in the table is clicked, a popup form appears, allowing the user to input the actual (physical) stock and add notes regarding the stock. As the user enters the actual stock value, the system automatically calculates and displays the difference between the actual stock and the stock recorded in the system. This feature helps ensure accurate stock reconciliation and provides clear documentation of discrepancies, supporting better decision-making and inventory control within the pharmacy. The UI design supports efficient, real-time stock auditing by streamlining data entry and enhancing transparency.

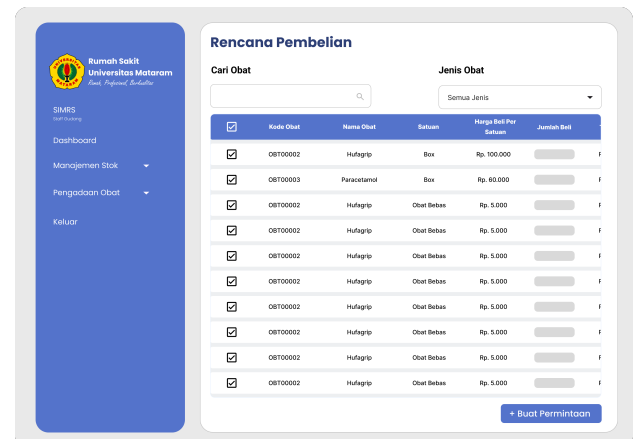


Fig. 11. Purchase Planning Page of Warehouse Staff

The UI shown in Figure 11 shows the user interface (UI) of the "Purchase Planning" page in the pharmacy stock management module of the Hospital Management Information System (SIMRS) at Mataram University. At the top, users can search for medicines by name and filter by drug type. Below that, a table lists medicine details including the drug code, name, unit (e.g., box), type (e.g., over-the-counter), unit price, and a field to enter the quantity to be ordered. Each item includes a checkbox, allowing users to select multiple drugs to include in a single purchase request. Once selections are made and quantities are specified, users can click the "+Buat Permintaan" (Create Request) button to generate a purchase request. This interface is designed to streamline the drug procurement process by making it easy to search, filter, and select medications for restocking, while also providing transparency on pricing and purchase planning in one view. After clicking the "Create Request" button, a popup form will appear as shown in Figure 12.

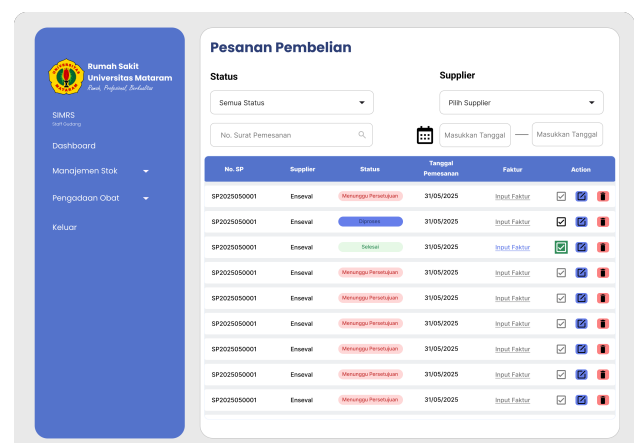


Fig. 12. Purchase Order Page of Warehouse Staff

The UI shown in Figure 11 shows the user interface (UI) of the Purchase Order page in the pharmacy stock management module of the Hospital Management Information System (SIMRS) at Mataram University. The UI is designed to facilitate the monitoring and management of the medication procurement process from suppliers.

Several key filtering elements are provided, including filters by *Status*, *Supplier*, *Purchase Order Number*, and *Order Date*, to help users quickly search for specific orders. The main table displays a list of purchase orders, including information such as the purchase order number (No. SP), supplier name, order status (e.g., *Pending Approval*, *In Process*, and *Completed*), order date, and an *Invoice Input* feature. The *Action* column includes icons for viewing details, approving, or deleting the order. The interface is structured and concise, making it easy for users to take direct action on available data, thereby supporting the warehouse staff's efficiency in managing the procurement of medicines.

E. Evaluating Architecture

To ensure that the system blueprint accurately reflects user needs and operational workflows, activity diagram, class diagram, component diagram, and sequence diagram was mapped back to the functional requirements collected during the analysis phase. This traceability confirms that all user tasks, such as inventory control, prescription verification, and purchase planning, are represented in the system architecture. Additionally, the modular design was reviewed to ensure scalability and ease of integration with other hospital module services. The architecture adheres to relevant regulations, including Ministry of Health Regulation No. 58/2014 on pharmaceutical services. While this evaluation remains at the design level, it ensures that the system blueprint is logically consistent, compliant, and ready for future implementation.

It is important to note that while the ERD, activity diagram, component diagram, class diagram, and sequence diagram are part of the system design documentation, the evaluation phase of this study specifically targets the user interface layer through usability testing using the UEQ method. The UML models serve as internal design references and are not subject to direct user evaluation.

The user experience (UX) evaluation of the hospital management information system pharmacy module at UNRAM Hospital was conducted by distributing questionnaires through Google Forms to the pharmacy installation staff at UNRAM Hospital, consisting of 5 pharmacists, 15 pharmacy assistants, and 15 pharmacy warehouse staff. From all the questionnaires distributed, a total of 10 respondents filled out the questionnaire with the following distribution: 2 pharmacists, 4 pharmacy assistants, and 4 warehouse staff. The rest of the staff chose not to fill the survey. Most of the staff targeted for the survey had busy work schedules within the hospital, and therefore did not have enough free time to complete the evaluation form despite being given prior notice.

The data obtained from the questionnaires was entered into the UEQ data analysis table, then transformed by calculating the values obtained from the User Experience Evaluation on the Information System using the User Experience Questionnaire. The transformed data was then processed to obtain the average value for each UEQ aspect. The UEQ results can be seen in Table 3.

TABLE 3. UEQ Evaluation Result

	UEQ Score	UX Aspects	UEQ Score
Attractiveness	1,63	Attractiveness	1,633
Pragmatic Quality	1,52	Perspicuity	1,625
		Efficiency	1,500
		Dependability	1,425
Hedonic Quality	1,56	Stimulation	1,675
		Novelty	1,450

As shown in Table 3, the UEQ (User Experience Questionnaire) evaluation results, the pharmacy module of UNRAM Hospital's management information system demonstrates moderately positive user experience across all measured aspects. The system performs best in the Stimulation aspect (1.68), indicating that users find the system engaging and motivating to use, followed by Attractiveness and Perspicuity (both at 1.63), suggesting users have a positive overall impression of the system and find it easy to understand and learn. The Hedonic Quality score (1.56) reflects moderate satisfaction with the system's aesthetic and experiential qualities, while the Pragmatic Quality score (1.52) indicates adequate usability in practical terms. The Efficiency score (1.50) shows that users can accomplish their tasks with reasonable effectiveness, though there's room for improvement in task completion speed. The system shows the lowest performance in Novelty (1.45) and Dependability (1.43), indicating that users may find the interface somewhat conventional and experience occasional concerns about system reliability. While all scores fall within the positive range, the relatively modest values (all below 1.7) suggest that there are opportunities for enhancement across all UX dimensions, particularly in improving system reliability and introducing more innovative interface elements to achieve a more optimal user experience.

Each average value from the UEQ calculation is compared with the benchmark values found in Table 3 to understand the meaning of each value. The results of the UEQ scale measurement compared with the benchmark values can be seen in Figure 10 below.

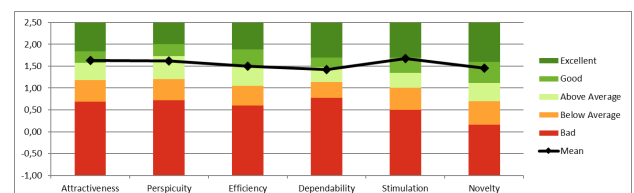


Fig. 13. Results of Comparison with Benchmark Scale

Based on the UEQ benchmark comparison chart, the pharmacy module of UNRAM Hospital's management information system shows consistently "Good" performance across all six user experience dimensions. All measured aspects - Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation, and Novelty - fall within the

"Good" category range (indicated by the darker green band), with mean values (black line with diamonds) hovering around 1.4-1.7. This indicates that the users find the system engaging, while Dependability and Novelty show slightly lower but still good performance. The consistent placement of all dimensions within the "Good" benchmark category indicates a well-balanced system that meets user expectations across pragmatic aspects (efficiency, perspicuity, dependability) and hedonic aspects (stimulation, novelty, attractiveness). However, none of the dimensions reach the "Excellent" category, suggesting there is still room for improvement to achieve outstanding user experience performance.

V. CONCLUSIONS AND SUGGESTIONS

A. Conclusions

This study designed a pharmacy management information system for the University of Mataram Hospital using the Total Architecture Synthesis (TAS) method and User-Centered Design (UCD) approach. TAS guided the structured development of system architecture through stages such as requirement analysis, business process modeling, and technical design. UCD involved end-users throughout the design process to ensure the interface matched their real needs and workflows. The combination of both methods resulted in a comprehensive system blueprint that not only supports robust technical implementation but also enhances user acceptance and operational practicality. This approach can serve as a reference for future development of hospital information systems, particularly in complex and user-dependent environments like pharmacy installations. The resulting system integrates key features such as Master Data Management, Inventory Management, and Patient Item Services, supported by complete architectural documentation and UI/UX prototypes. Usability testing with the User Experience Questionnaire (UEQ) yielded "Good" results across all dimensions, with Stimulation scoring the highest (1.68), indicating that the system is both functionally robust and user-friendly.

B. Suggestions

To achieve "Excellent" user experience ratings, improvements are recommended in three key areas: Dependability (1.43) through enhanced data validation, error handling, and system recovery mechanisms; Novelty (1.45) via innovative interface design with modern visual elements and interactive components; and Efficiency (1.50) by simplifying user workflows and adding quick access features.

REFERENCES

- [1] HAKIM, Dimara Kusuma. Meningkatkan Efisiensi dan Efektivitas Pelayanan Rumah Sakit melalui Pelatihan SIMRS di Rumah Sakit Bunga Bangsa Medika: Improving the Efficiency and Effectiveness of Hospital Services through SIMRS Training at Bunga Bangsa Medika Hospital. *Jurnal Pengabdian Kepada Masyarakat (JUDIKA)*, 2024, 5.2: 19-27.
- [2] KRISTANTI, Yuliana Erma; AIN, Ratu Qurroh. Sistem informasi manajemen rumah sakit: Literature review. *Muhammadiyah Public Health Journal*, 2021, 1.2: 179-193.
- [3] MEGAWATY, Megawaty. Penerapan Model Pendekatan Total Architecture Sintesis dalam pembangunan Sistem Laboratory Bumi Insani. *Penerapan Model Pendekatan Total Architecture Sintesis dalam pembangunan Sistem Laboratory Bumi Insani Megawaty*, 2022.
- [4] C. Ravelino, Y. Alfa Susetyo, and K. Satya Wacana, "Perancangan UI/UX untuk Aplikasi Bank Jago menggunakan Metode User Centered Design," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 7, no. 1, p. 2023.
- [5] S. Rasio Henim and R. Perdana Sari, "Jurnal Politeknik Caltex Riau Evaluasi User Experience Sistem Informasi Akademik Mahasiswa pada Perguruan Tinggi Menggunakan User Experience Questionnaire," 2020.
- [6] MURSALIN, Rasyad Alief, et al. Pengembangan Sistem Informasi Manajemen Rumah Sakit Menggunakan Framework PHP. *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, 2023, 2.2: 93-99.
- [7] D. Wira, T. Putra, and R. Andriani, "Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD," vol. 7, no. 1, 2019.
- [8] "PERATURAN MENTERI KESEHATAN REPUBLIK INDONESIA No. 58, Tahun 2014."
- [9] B. Unhelkar, "Software Engineering with UML.", Auerbach Publications, 2020.
- [10] A. Bayu Saputra, A. Cahyono, U. K. Jenderal Achmad Yani Yogyakarta Jl Siliwangi, and R. Barat Banyuraden Gamping Sleman Yogyakarta, "Analisis dan Perancangan Laboratory Information Management System (LIMS) Menggunakan Metode Total Architecture Synthesis (TAS)," MEI, 2020.
- [11] Denierlansyah, Denierlansyah. "Sistem Informasi Pengolahan Data Kecelakaan Berbasis Web Menggunakan Metode Total Architecture Syntesis. *Sistem Informasi Pengolahan Data Kecelakaan Berbasis Web Menggunakan Metode Total Architecture Syntesis*", 2022.
- [12] KARIM, Windy; KATILI, Muhammad Rifai; PADIKU, Indhitya R. Pengembangan Sistem Informasi Persediaan Obat Berbasis Web Menggunakan Metode Prototype Di Instalasi Farmasi Rumah Sakit Otanaha Kota Gorontalo. *Diffusion: Journal of Systems and Information Technology*, 2025, 5.1: 55-65.
- [13] Abbasi, R., Farzandipour, M., Rahimi, H. et al. "A heuristic evaluation of a pharmacy surveillance information system." *BMC Med Inform Decis*, 2024.
- [14] SANGADJI, Ahmad Rizqa Habibi Putra. *TA: Prototype UI/UX Sistem Informasi Manajemen Rumah Sakit (SIMRS) Berbasis Mobile dengan Metode User Centered Design*. 2025. PhD Thesis. Universitas Dinamika.
- [15] PANE, Maya Saufinah, et al. Sistem Informasi Manajemen Rumah Sakit (SIMRS) Untuk Meningkatkan Mutu Pelayanan Kesehatan Di Indonesia. *Detector: Jurnal Inovasi Riset Ilmu Kesehatan*, 2023, 1.3: 01-14.