Jurnal Sistematik
INSTITUT TEKNOLOGI GARUT

Jurnal Sistem Informasi, Teknologi Informasi dan Komunikasi
DOI: 10.33364/sistematik/v.1-1.2405

Application of Extreme Programming Method for Digitization of the Integrated Drug Prescribing Process Patient Registration of Bunda Alya Clinic

Ridwan Setiawan[1], Lulu Putri Novianti[2]
[1,2]Department of Informatics, Institut Teknologi Garut, Indonesia

***email:** ridwan.setiawan@itg.ac.id

Article Info
Submitted: May 1, 2025
Received: May 14, 2025
Published: May 31, 2025

Keywords:
Extreme Programming;
Patient Data Integration;
Prescription Electronic Medicine;
Web Application.

ABSTRACT

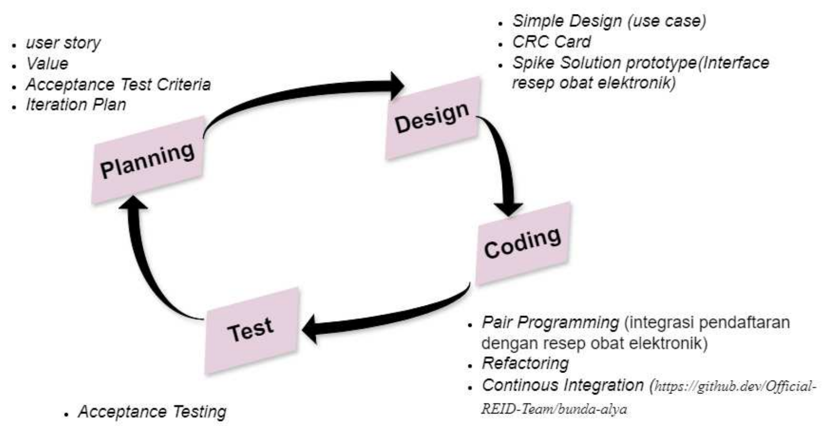
Digital transformation in the healthcare sector is driving the need for systems that can improve the efficiency and accuracy of clinic services. This research develops a web-based electronic drug prescription application that is integrated with the patient registration system, with an Extreme Programming (XP) approach. The system is designed to overcome manual process constraints in the flow of medical services, from registration to printing drug recommendations. The test results showed that the application was able to reduce the service time from an average of 8 minutes to 4 minutes per patient. The User Acceptance Test test of 13 respondents showed a satisfaction rate of 95%. These findings confirm that the systems built make a significant contribution to improving the efficiency of the clinic's work and minimizing the risk of medical data recording errors. This application is expected to be an applicative and sustainable digital solution in the context of primary health services.

1. INTRODUCTION

Digital transformation in the health sector has become an essential need to improve the quality and efficiency of medical services. One aspect that still relies heavily on manual processes is the patient registration and drug prescribing system in small and medium clinics. Studies show that about 25% of [1], [2][3]medication errors occur due to incomplete patient information, such as age or disease history, in manual prescriptions. This poses a risk to the accuracy of dose, the effectiveness of treatment, and the safety of patients.[4]. The Bunda Alya Clinic in Garut Regency is a real example of this challenge. The medical service process at this clinic is still carried out manually, starting from registration, recording doctor's diagnosis, to prescribing and labeling drugs by pharmacists. These system limitations cause services to be slow, prone to data loss, and require inefficient cross-section communication. This situation highlights the need for a digital system that not only simplifies the process, but also ensures the accuracy and integration of patient data across all stages of clinical services.[5], [6]. Various previous studies have sought digitalization in the context of clinical services. Research developed a cash-connected electronic prescription app to cut patient payment waiting times. The study proposes a cloud-based system and QR code to avoid misreading doctors' handwriting. While researching and applying the Extreme Programming (XP) methodology to develop a medical record system with results that are adaptive to user needs. However, most of these studies only touch part of the clinical service chain without integrating the flow of registration, doctor diagnosis, prescription, and drug recap and labeling in one unified system.[7][8][9][10]. The research gap that this study aims to fill lies in the development of a comprehensive health information system. It takes an approach that is able to integrate the entire service process starting from patient registration, filling out diagnoses, giving prescriptions, to printing recommendations for the use of drugs in one web-based platform. Using the Extreme Programming methodology, the system is designed to respond to user needs iteratively and flexibly, with the main goal of improving operational efficiency, reducing the risk of errors, and speeding up the service process in the clinic. This research aims to design and implement a web-based electronic drug prescription application that is integrated with the patient registration system at the Bunda Alya Clinic. The application is expected to be an end-to-end digital solution that supports the comprehensive and sustainable digitization of clinical services.

2. RESEARCH METHODOLOGY

This research uses the XP approach, one of the agile methodologies that emphasizes software development iteratively, quickly, and responsively to changing user needs. This methodology was chosen because it is able to produce a stable and tested system through intensive collaboration between developers and end users. The XP stages used in this study consist of: planning, design, coding, and testing. The stages of the research method are shown in Picture1.[11] - [15]



Picture1. Extreme Programming (XP) Methodology Flow

Based on Picture1, the flow of the research method with XP Programming, the explanation of each stage is as follows:

1) Planning. At this stage, the identification of system needs is carried out through interviews and observations of business processes in the prescription implementation process at the Bunda Alya Clinic. This need is formulated in the form of [16]user stories that describe the desired features from the user's perspective. Each User Story is accompanied by a value and acceptance test criteria. The results of this stage are presented in the form of a table that lists the user code, a description of the requirements, important values, and the test scenario. In addition, an iteration plan is prepared to determine feature development priorities, categorized into high, medium, and low scales.
2) Design. The design stage in XP prioritizes the principle of Keep It Simple (KIS). The design is kept to a minimum while still meeting the functional needs of the system. Some of the artifacts developed at this stage include: (1) [17]Use case diagrams depicting the relationship between system actors and their functions; (2) Class diagrams show the class structure in the system, including attributes and relationships between entities; (3) CRC Card (Class-Responsibility-Collaborator) which is used to define the responsibilities of each class and its collaboration in the context of object orientation; and (4) Wireframe Interface (Spike solution Prototype) which is used to display the system display as an initial validation tool by the user.
3) Coding. The coding is done based on a pre-made design. This stage implements XP practices such as pair programming, where two developers work on one code collaboratively to improve quality and reduce errors. In addition, periodic [14], [18]refactoring is carried out to simplify the code structure without changing functionality, as well as continuous integration that ensures that the code is always in a condition that can be run and tested.
4) Testing, this stage is for testing the features on the application so that there are no errors (Error), ensure the software or system runs as expected, meets set specifications, and meets user needs [19].

3. RESULTS AND DISCUSSION

In this part of the results and discussion, the results of the implementation of each stage in the Extreme Programming (XP) approach used in system development are presented. These stages include Planning, Design, Coding, and Testing. The main focus is on the interconnectedness between user needs and the technical solutions being designed, as well as how the system is tested and validated against the functional needs of the clinic.

3.1 Planning

The planning stage is the initial phase in the user-oriented system development process. At this stage, the identification of system needs is carried out based on interviews with the main users, namely registration admins, doctors, and pharmacists. The result of this stage is the documentation of needs in the form of user stories that contain actors, functional needs, value values, and acceptance criteria. The user stories that have been identified then become the basis for the preparation of iteration plans and iterative system development flows. The output of planning activities is in the form of functional specifications of the system in the form of narratives that are classified based on user roles. This specification describes the interactions, information needs, and work processes supported by the built-in electronic drug prescription information system. Table 1 presents a comprehensive mapping of user stories and the actors involved, the functional value provided by each user story (value), and the system acceptance criteria that are the basis for validation during the testing process.

Table 1. Mapping User Stories, Actors, Values, and Acceptance Criteria

Code	About Shawn Stevens	Actor	Value	Acceptance Criteria
US1	Users can log in to the system according to their role	Admin, Doctor, Pharmacist	Secure authentication and access authorization	The system directs users to the role-based dashboard after username and password validation
US2	Admins can add new patient data	Admin	Ease of patient data entry	The input form receives the name, NIK, date of birth, and address; Data stored in a database
US3	Admins can manage patient data (view, edit, delete)	Admin	Data management flexibility	The system displays a list of patients and provides change and delete functions that work as expected
US4	Admin can fill out the patient registration form	Admin	Efficiency of the registration process	The filled in data is stored and connected directly to the patient's medical records
US5	The system can display a list of patients who have been registered	Admin	Enrollment data visibility and control	The admin page displays active patient data in real-time
US6	The doctor can input the patient's diagnosis results	Doctor	Documentation of the audit results	Diagnoses are stored in databases and can be accessed by doctors and pharmacists
US7	Doctors can access the patient's medical records	Doctor	Medical decision-making support	Integrated medical records appear in full on the doctor's page
US8	Doctors may prescribe medication after diagnosis	Doctor	Appropriate treatment recommendations	The active drug input form is as per the diagnosis, and the data is stored in the prescription system
US9	Pharmacists can recap prescriptions from doctors	Pharmacist	Access precise recipe information	The recap page displays prescription data according to the patients served
US10	Pharmacists can download prescription recap data	Pharmacist	Local documentation of patient prescriptions	The system provides a data export button in downloadable .xls format
US11	Pharmacists can fill in the recommendations for the use of drugs	Pharmacist	Patient drug use education	Consumption instruction inputs are available and stored according to the relevant prescription
US12	Pharmacists can print labels recommended for use	Pharmacist	Effectiveness of drug distribution	The system generates .pdf files that can be printed and pasted on drug packaging

As shown in Table 1, it shows that the needs of users have been systematically classified based on their respective roles, so that the developed system actually accommodates real workflows in the field. For example, in US1 to US5, the functional needs of the admin are more focused on managing patient data and enrollment.

While US6 to US8 focuses on the process of diagnosis and prescribing by doctors, while US9 to US12 leads to the management of prescription information and drug use labels by pharmacists.

3.2 Design

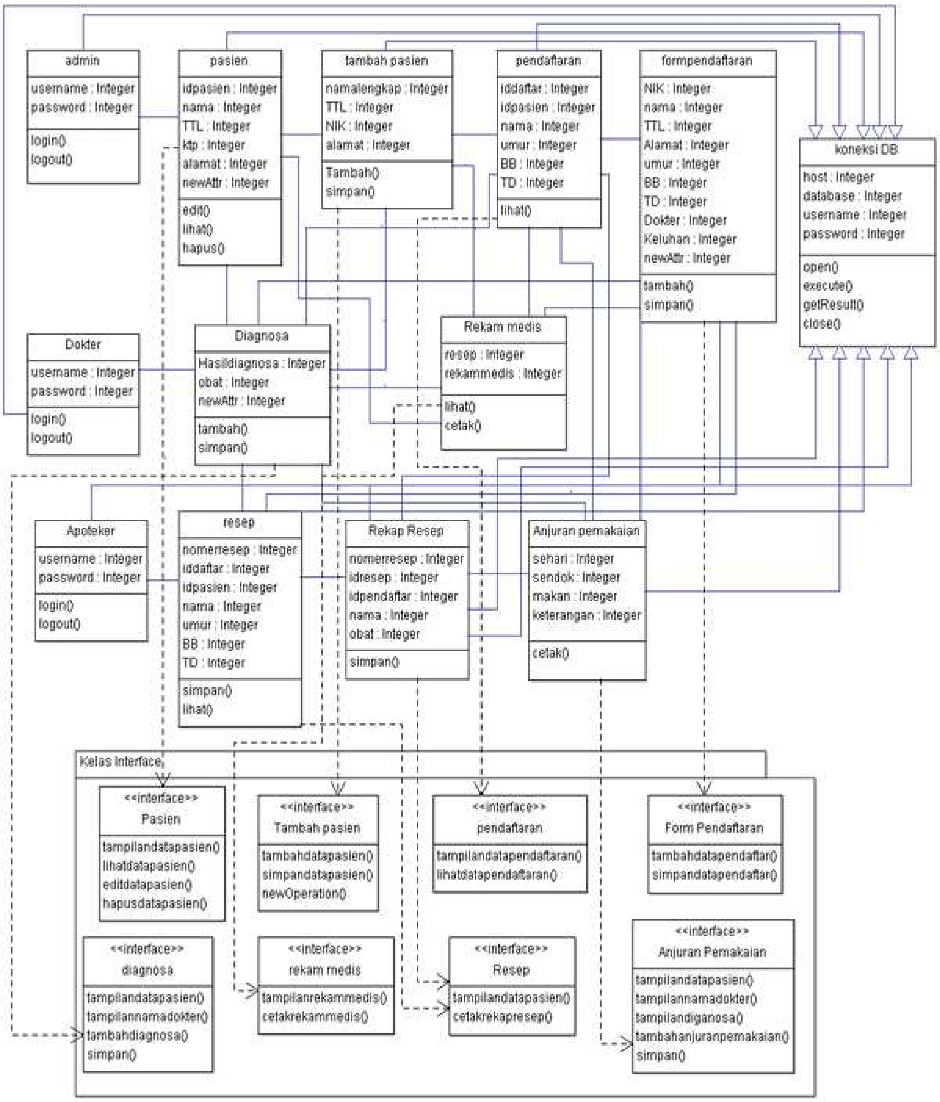
The design stage is an important process in software development that aims to transform the functional needs of the system into a form of technical visualization that can be used as a reference for implementation. In this study, the design was carried out with an object-oriented approach using Unified Modeling Language (UML) and Class Responsibility Collaborator (CRC Card) cards to clarify the responsibilities of each system component. The design of the system also produces a user interface mockup model that is intended to reflect the actual needs of each user role.

1) Use Case Diagram. To model the interaction scenario between actors and systems, a Use Case Diagram as shown in Picture2 is used.

Picture2. Use Case Diagram of Electronic Drug Prescribing System

Based on Picture2, the system interacts with three main actors, namely Admin, Doctor, and Pharmacist. Admins are in charge of patient data management and enrolment. Doctors use the system to record the results of the diagnosis and prescribe medications. Pharmacists play a role in the input of drug use recommendations, as well as the printing of prescription labels based on the information that has been provided by the doctor. The use cases described include login, patient data management, diagnostic input, prescription administration, prescription recap, and printing of drug recommendations. This diagram ensures that all operational needs have been systematically mapped.

2) Class Diagram, To model the logical structure and relationships between entities, use Class Diagrams as shown in Picture3.



Picture3. Electronic Drug Prescribing System Diagram Class

Based on Picture3. This class of electronic drug prescription system diagrams is designed to represent the internal structure of an object-based system consisting of several main entities: system users, medical service processes, drug prescription management, and interface support components. This diagram presents relationships between classes, attributes, and methods that reflect real processes in web-based clinical services. On the user side, there are three main classes, namely admin, doctor, and pharmacist, each of which has authentication attributes in the form of a username and password, as well as the login() and logout() functions. These three actors interact with other classes according to their roles in the clinical process. Admins play a role in managing patient data and registration, doctors are responsible for filling out diagnoses and prescriptions, while pharmacists manage medication usage information and printing prescription labels. Patient classes store patient identity information such as patient id, name, TTL, address, and newArt, and are equipped with data management functions such as edit(), view(), and delete(). The process of adding patient data is facilitated by the add-patient class, which presents a complete input form with the add() and save() methods. This data is then used in the registration process, which manages the patient's registration information and initial condition (weight, blood pressure, age) through the view() function. The diagnosis class serves to record the results of examinations carried out by doctors on patients. Its attributes include diagnostic results, drugs, and patient ids, while its functions include add() and save(). The results of this diagnosis will be the basis for making prescriptions, which contain data such as prescription number, patient ID, age, medication, as well as blood pressure and weight values. This process supports the integration of medical outcomes with treatment therapy.

Prescription recapitulation is managed in the prescription recap class, which organizes data for drug documentation and audit purposes. Meanwhile, the recommended use class allows pharmacists to add information on drug use such as frequency of consumption, meal time, number of spoons, and additional information. The print() function in this class is used to generate labels in print-ready format (PDF).

The system also includes a class of medical records that serve as a historical repository of all of the patient's clinical activities, such as the results of examinations and previous prescriptions. Its functions include view() and print() to facilitate long-term tracking of medical information. To ensure the separation of presentation logic from business logic, the system uses a series of interface classes such as <<patient interface>>, <<interface>>diagnosis, <<interface>> prescription, and <<interface>> usage recommendations. These classes are in charge of displaying data to users, receiving input, and connecting to the appropriate modules. This design is in line with the principles of MVC (Model-View-Controller), where the display logic and processes are controlled through a modular interface. Finally, connections to the database system are governed through the connectionDB class, which has attributes such as host, database, username, and password, as well as the open(), execute(), getResult(), and close() methods. This class is an important foundation for the data transaction process across all system modules. Overall, this class diagram shows a coherent and modular system architecture. Each class is designed with a single responsibility, allowing for efficient integration between modules, and supporting future maintenance and development of the system.

3) CRC Card. To strengthen the design of object-oriented systems, the Class Responsibility Collaborator (CRC Card) approach is used as a conceptual technique in defining the structure of responsibility and collaboration between classes. CRC Card is a design analysis method that focuses on three main elements: Class, which is the entity in the system; Responsibility, which is the main task or function of the class; and Collaborators, which are other classes that are involved in assisting or interacting with the class to complete its responsibilities.
In this study, CRC Cards are arranged based on all entities in the diagram class, which includes user actor classes, business process classes, and supporting components such as database connections and input forms. Each class is identified as having specific primary responsibilities and only collaborates in case of data dependencies or process flows between classes.
For the sake of academic presentation in journal articles, the following Table 2 is presented as a form of narrative summary of the CRC Card that has been analysed. It should be noted that Table 2 is not a literal representation of the CRC card, but rather a summary describing the elements of the CRC in one unified tabular format for presentation efficiency and clarity of function mapping between classes.

Table 2. CRC Card Combined Electronic Drug Prescription System

CRC Card Name	Responsibilities	Collaborators
Admin	Authentication as an admin, managing patient data and enrolment	Patient, Registration, AddPatient
Doctor	Authentication as a doctor, input diagnoses and prescriptions	Diagnosis, Prescription, Medical Records
Pharmacist	Authenticate as a pharmacist, input usage recommendations and print labels	RecommendedUsage, Recipes, RecapRecipes
E-Mail	General structure of user authentication	SystemAuth
Patient	Store and display patient identity data	Admin, Registration, Medical Records
AddPatient	New patient data input form	Patient, Admin
Registration	Record the patient's initial medical registration and status	Registration Form, Patient
FormRegistration	Provide registration input forms for admins	Admin, Registration

CRC Card Name	Responsibilities	Collaborators
Diagnosis	Record the results of the doctor's examination and connect it to the prescription	Doctor, Patient, Prescription
Recipe	Stores prescription information provided by the doctor, including medication data and patient condition	Diagnosis, Doctor, Medication, Recommended UseUse
Medicine	Storing medication and dosage list data	Recipe
RecapRecipes	Save and display prescription recaps for pharmacists	Prescription, Pharmacist
RecommendedUsage	Fill out the medication recommendations and print labels	Pharmacist, Prescription
Medical Records	Storing and printing data of patients' historical medical records	Patient, Diagnosis, Prescription
KoneksiDB	Execute database operations (open, execute, getResult, close)	All data storage classes

Based on the information in Table 2, it can be seen that each class in the system has focused responsibilities. For example, the Prescription class not only acts as a container for drug data, but also serves as a link between diagnostic results and recommended use, so its role is very strategic in the flow of this digital clinic system. While the Recommended Use class collaborates directly with the Prescription and Pharmacist, and holds the final function before the information is given to the patient in the form of drug labels.

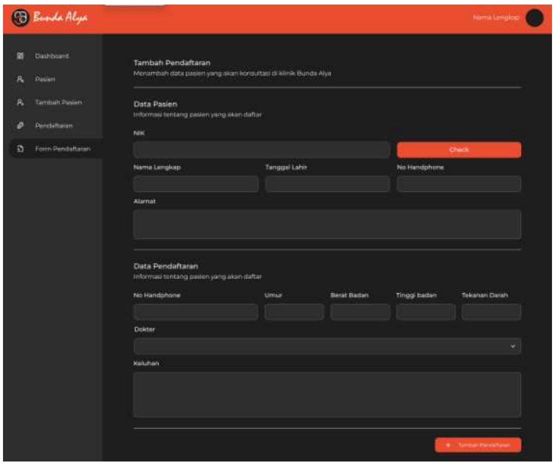
4) Spike solution Prototype. As part of the Extreme Programming (XP) practice, the Spike solution was used in this study to test the technical and functional feasibility of the core features of the system before full implementation was carried out. The main objective is to evaluate the alternative design and validity of the frontend and backend technologies to be used, especially the integration between input forms, patient data storage, and the printing process of drug use recommendations. The spike solution was carried out in the form of initial interface prototyping using Figma, as well as proof-of-concept data processing on the NextJS and ReactJS frameworks to ensure responsiveness and compatibility of web displays. The results of this spike are then used as a reference in the development of the final interface and data storage structure.

3.3 Coding

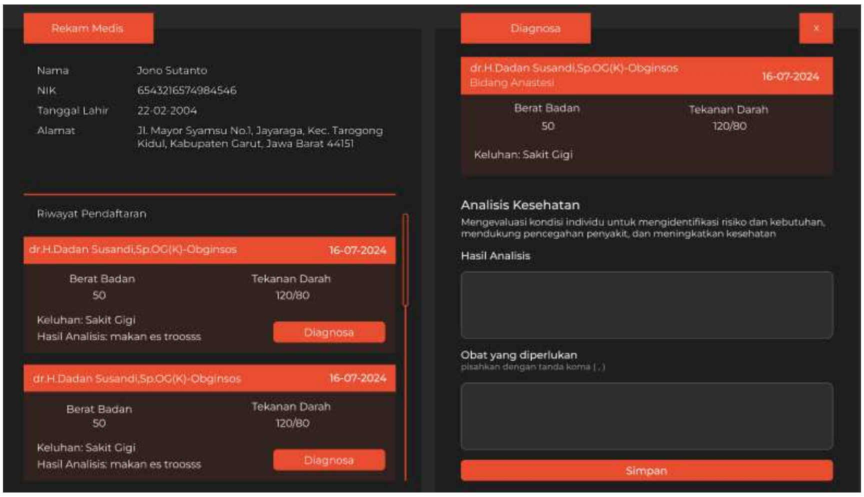
The coding stage is the implementation of a system design that has been designed beforehand. At this stage, all the components that have been identified in the diagram class, CRC Card, and interface mockup are transformed into real program code using a component-based modular approach. The coding process is carried out by applying the principle of incremental development according to the Extreme Programming (XP) approach, where each iteration focuses on specific functional development based on the priority order of user needs.

The development of this system was carried out using NextJS and ReactJS technology for the frontend side, as well as Node.js and MySQL for data management on the backend side. This framework was chosen because it supports the development of web-based applications that are reactive, responsive, and support real-time component integration.

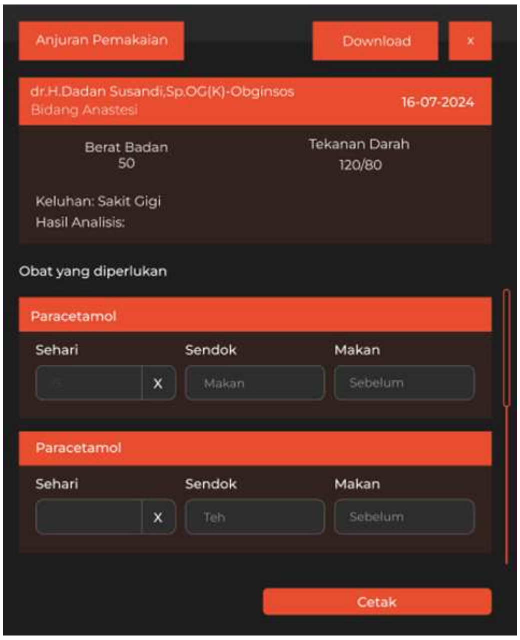
The coding structure is divided into several main modules according to the user's role, namely the admin module, the doctor module, and the pharmacist module. Each module includes a login page, a main dashboard, and an interactive form that supports related business processes. The interface components are built on prototypes that have been validated at the spike solution stage.



Picture4. Implementation of Patient Registration Form Page (Admin)



Picture5. Implementation of the Diagnosis and Prescription Page (Doctor)



Picture6. Implementation of Recommendation for Use Page and Prescription Label (Pharmacist)

Based on Picture4, the admin page allows patient data input and registration through forms that are directly connected to the database. Validation is done on a client-side and server-side basis to ensure data accuracy. Picture5 shows the physician interface integrating the diagnosis form and prescription filling. This component utilizes state management in ReactJS to maintain active synchronization of patient data and diagnostics. Picture6 shows a page for pharmacists that includes an input form for use as well as a label printing function in PDF format. The print process is done through the integration of external libraries such as react-to-print or html2pdf, ensuring compatibility for physical pharmaceutical documentation.

This coding process is also accompanied by writing API functions on the backend side to handle CRUD (Create, Read, Update, Delete) operations on patient data, prescriptions, and drug recommendations. Each request from the frontend is sent via HTTP Request and processed by the endpoint on NextJS API routing, then forwarded to the MySQL database with a secure and efficient structured query.

The development architecture is structured with the principle of separation of concern, where display logic, business logic, and data storage logic are separated to support code readability and scalability. Code reusability and componentization practices are also implemented to speed up subsequent iterations and reduce code redundancy.

3.4 Testing

Testing by determining respondents first using User Acceptance Test conducted to 13 respondents with 7 questions with a questionnaire using Likert Scale[20] with the questionnaire using the Likert Scale, namely strongly agree (SS), agree (S), strongly agree (CS), disagree (TS) and strongly disagree (STS).

Table 3. User Acceptance

Yes	Question	Responses				
		ST 5	**S 4**	**C 3**	**TS 2**	**STS 1**
1	Is this e-drug prescription app easy to use?	10	3			
2	Is Display For app already interesting?	11	2			
3	Are in-app flows easy to understand?	9	4			
4	Does this app help in the healthcare process?	10	3			
5	Is this application useful in health services?	9	4			
6	Does the app run well?	11	1	1		
7	Do you agree with this application?	9	4			
	Total	69	21	1		

a. Calculating the Total Score based on respondents' answers
SS = 69 x 5 = 345
S = 21 x 4 = 84
N = 1 x 3 = 3
CS= 0 x 0 = 0
TS= 0 x 0 = 0
Total Value = SS + S + CS + TS + STS
Total Score = 345 + 84 +1 = 432

b. Finding an X value
X = 5 x (SS+S+N+CS+TS)
X = 5 x 91= 455

c. Find a percentage value
The value of X is used as a divisor because the question asked is a positive question.
Index formula = $\left(\frac{\text{Total Value}}{X}\right)100\%$

$$\text{Formula Index} = \left(\frac{432}{455}\right) 100\% = 95\% \text{ (Strongly Agree)}$$

So it can present the results of the overall percentage by using the UAT testing method, the criteria are very agreeable with the average of 95 percent. With these results, it can be concluded that users feel very satisfied with this electronic drug prescription application.

3.5 Discussion

The results of the implementation of the electronic drug prescription system integrated with patient registration show success in answering the main problems that occur at the Bunda Alya Clinic, especially in terms of the efficiency of the service process and the accuracy of recording medical information. Based on the results of the User Acceptance Test (UAT) of 13 respondents, the system obtained a satisfaction rate of 95%, indicating that the majority of users strongly agreed that the application is easy to use, supports clinical workflows, and is suitable for continuous implementation. In terms of efficiency, the service time that was originally an average of 8 minutes per patient can be reduced to 4 minutes after the system is implemented. This is in line with the findings that developed an electronic prescription application connected to the cashier and was able to cut patient transaction time. However, the system developed in this study has the advantage of providing comprehensive integration from registration, diagnosis, prescription, to drug label printing, not only limited to the final stage of the service process.[7]

From a technical point of view, the use of the Extreme Programming (XP) methodology has proven to be effective in producing a system that is responsive to user needs, in line with studies and which shows the success of XP in the development of medical record systems. However, unlike these studies that focus on medical data management only, this study has succeeded in integrating the entire clinical service cycle into one web-based digital system.[9][10]

In addition, the system design approach involving user stories, CRC Cards, as well as UML modeling such as use case diagrams and class diagrams, has been proven to help the development team in translating functional requirements into modular application modules that can be tested independently. The refactoring and componentization process applied during coding also strengthens the quality of the system and facilitates the integration process of interfaces and backends. Compared to cloud-based systems and QR codes as developed by, this system provides added value through automatic printing of drug recommendation labels, which is very important to ensure patients get clear medication usage information. This feature also supports more systematic patient education, which has not been the main focus in previous research.[8] Thus, the system developed in this study contributes as a comprehensive digital solution for clinics with manual workflows, who want to adopt web-based electronic systems without losing the workflow that has been running before. This makes the system not only technically innovative, but also relevant and applicable in the context of primary healthcare at the local level.

4. CONCLUSION

This research has succeeded in developing a web-based electronic drug prescription application that is integrated with the patient registration system at Bunda Alya Clinic. The development process is carried out with an Extreme Programming (XP) approach, which allows the system to be built iteratively and adaptively to the needs of the user. The resulting application is able to digitize the entire clinical service process, from patient registration, diagnostic input, drug prescription, to printing recommended use in the form of labels. The results of the test using the User Acceptance Test showed a user satisfaction rate of 95%, as well as a reduction in service time from an average of 8 minutes to 4 minutes per patient. These findings show that the system is able to improve efficiency, reduce manual recording errors, and speed up the flow of medical services. The main contribution of this research is the preparation of an integrated clinical information system, with an easy-to-use interface that can be implemented directly in the context of primary health services. This system is expected to be an applicable and sustainable solution for the digitization of clinic services in the future.

REFERENCE

[1] M. R. Fauzi, S. Saimi, and F. Fathoni, “Tantangan dan Solusi Administrasi Kesehatan di Era Digital (Tinjauan Literature Review atas Implementasi Teknologi),” AL-MIKRAJ Jurnal Studi Islam Dan Humaniora (E-ISSN 2745-4584), vol. 5, no. 01, pp. 1093–1103, 2024.

[2] M. Wulandari, T. Novriyanti, P. Purwadhi, and Y. R. Widjaja, “Implementasi Strategi Transformasi Digital dalam Meningkatkan Kualitas Pelayanan di Rumah Sakit: Studi Kualitatif,” Innovative: Journal Of Social Science Research, vol. 5, no. 1, pp. 1415–1427, 2025.

[3] T. Maalangen, G. Citraningtyas, and W. I. Wiyono, “Identifikasi Medication Error Pada Resep Pasien Poli Interna Di Instalasi Farmasi Rumah Sakit Bhayangkara Tk. III Manado,” Pharmacon, vol. 8, no. 2, pp. 434–441, 2019.

[4] P. Gupta, A. Lachyan, P. Chandil, and M. Tamrakar, “Medication Error Audits: A Comprehensive Overview and Implications for Patient Safety,” Journal of the Epidemiology Foundation of India, vol. 2, no. 2, pp. 46–49, 2024.

[5] D. Maste, N. Patil, and M. Meena, “Review of Healthcare Database,” in 2020 International Conference on Convergence to Digital World-Quo Vadis (ICCDW), IEEE, 2020, pp. 1–6.

[6] K. N. B. A. Bakar, M. D. Nurpathama, and H. B. M. Hariri, “Clinic Management System,” Jurnal Inovasi Global, vol. 1, no. 2, pp. 179–191, 2023.

[7] H. Susilawati, T. A. Wiharso, and T. M. Hidayat, “Rancang Bangun Aplikasi Elektronik Resep yang Terhubung ke Kasir Menggunakan Website,” Jurnal Teknologi Informasi dan Terapan, vol. 7, no. 2, pp. 86–91, 2020.

[8] I. Saputra et al., “Sistem Berbasis Cloud Computing Untuk Identifikasi Resep Dokter ‘BarSep,’” Faktor Exacta, vol. 13, no. 4, pp. 232–242, 2021.

[9] D. Kurniadi, A. Mulyani, and D. A. N. Annisa, “Rancang Bangun Sistem Informasi Rekam Medis Pada Klinik Menggunakan Metode Extreme Programing,” Jurnal Algoritma, vol. 17, no. 2, pp. 440–451, 2020.

[10] E. B. Pratama and A. Hendini, “Implementasi Extreme Programming Pada Perancangan SIMRS (Sistem Informasi Manajemen Rumah Sakit),” Jurnal Khatulistiwa Informatika, vol. 10, no. 2, pp. 107–112, 2022.

[11] R. A. N. Putra, R. Kurniati, and E. Yumami, “Implementasi Extreme Programming untuk Sistem Jasa Pemesanan Design Website dan UI/UX,” SABER: Jurnal Teknik Informatika, Sains dan Ilmu Komunikasi, vol. 2, no. 4, pp. 202–226, 2024.

[12] A. Pambudi and W. Apriandari, “An Extreme Programming Approach for Instructor Performance Evaluation System Development,” Journal of Informatics Information System Software Engineering and Applications (INISTA), vol. 5, no. 2, pp. 126–135, 2023.

[13] I. Riani and R. Firdaus, “Implementasi Metode Extreme Programming dalam Sistem Informasi Manajemen Pemesanan Barang Online,” Bridge: Jurnal publikasi Sistem Informasi dan Telekomunikasi, vol. 2, no. 3, pp. 17–23, 2024.

[14] K. Pal, “eXtreme Programming, Agile Methodologies, Software Project Management, Customer Role, Rigorous Testing,” in Contemporary Challenges for Agile Project Management, IGI Global Scientific Publishing, 2022, pp. 163–182.

[15] D. Riskiyana, D. Sakethi, and R. Andrian, “Penerapan Metode Extreme Programming Pada Pengembangan Sistem Informasi Evaluasi Hafalan Al-Qur’an,” Jurnal Pepadun, vol. 2, no. 3, pp. 336–343, 2021.

[16] R. Setiawan and R. Maulana, “Sistem Informasi Sarana dan Prasarana Pendidikan Anak Usia Dini Menggunakan Metodologi Extreme Programming,” Jurnal Algoritma, vol. 17, no. 2, pp. 539–546, 2020.

[17] R. Putri, “Perancangan Sistem Informasi Desa Menggunakan Exteme Programming,” Journal of Information Technology, Software Engineering and Computer Science, vol. 2, no. 3, pp. 140–149, 2024.

[18] D. Wells, “Extreme Programming: A Gentle Introduction.” Accessed: Jul. 06, 2021. [Online]. Available: http://www.extremeprogramming.org/

[19] R. S. Pressman, Software engineering is software engineering. 2010. doi: 10.1049/ic:20040411.

[20] S. R. Widyastuti, “Pengembangan skala likert Untuk mengukur sikap terhadap penerapan penilaian autentik siswa sekolah menengah pertama,” Aswaja, vol. 3, no. 2, pp. 57–76, 2022.