International Journal Software Engineering and Computer Science (IJSECS)
5 (3), 2025, 1244-1259
Published Online December 2025 in IJSECS (http://www.journal.lembagakita.org/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v5i3.5354.

RESEARCH ARTICLE Open Access

Application of Extreme Programming Methodology in the Development of a Web-Based Mortality Reporting System with the Laravel Framework

Iin Inarotul Huda *
Health Information Management Study Program, Politeknik Piksi Ganesha, Bandung City, West Java Province, Indonesia.
Corresponding Email: iinhuda31@gmail.com.

Yuyun Yunengsih
Health Information Management Study Program, Politeknik Piksi Ganesha, Bandung City, West Java Province, Indonesia.

Falaah Abdussalaam
Health Information Management Study Program, Politeknik Piksi Ganesha, Bandung City, West Java Province, Indonesia.

Received: August 14, 2025; Accepted: November 15, 2025; Published: December 1, 2025.

Abstract: Hospital mortality data management directly influences service quality and strategic decision-making capabilities. While electronic medical records have become standard practice in healthcare facilities, existing systems frequently lack the adaptive features necessary for efficient mortality reporting. We address these limitations through the development of a web-based mortality reporting system using Extreme Programming (XP) methodology and the Laravel Framework. The development process engaged two primary user groups: nurses responsible for data entry and medical record officers managing validation and reporting. Black-box testing validated system functionality against operational requirements. Performance evaluation revealed substantial operational gains: patient data entry time decreased from 7-10 minutes to 2-3 minutes, duplicate validation transitioned from manual verification to automated real-time checking, and quarterly report generation reduced from 2-3 hours to 10-20 seconds. The XP methodology, particularly pair programming practices, accelerated defect resolution and enhanced system responsiveness to user feedback. These results demonstrate that iterative development approaches combined with modern web frameworks can fundamentally transform healthcare data management workflows, replacing error-prone manual processes with reliable, efficient digital solutions.

Keywords: Electronic Medical Records; Healthcare Information Systems; Mortality Reporting; Extreme Programming; Laravel Framework; Agile Development.

1. Introduction

Digital transformation has totally changed the way we deliver healthcare services. Technology is now the backbone of how modern medical practice operates. Healthcare institutions understand that technology helps clinical staff do their jobs better and more efficiently. Computer systems have moved from being simple data storage tools to advanced platforms that enhance performance, streamline workflows, and ensure accurate data is presented [1]. Medical records are the main source of information for both hospitals and patients, directly affecting the quality of services offered and the efficiency of operations within those facilities. Regulations require that medical records be kept both in written form and electronically to ensure complete documentation of patient information [2]. Healthcare facilities are responsible for processing information accurately and promptly so that accurate and actionable data can be provided to end users [3]. The implementation of electronic medical records covers the entire lifecycle of a patient from admission until discharge or referral or death with confidentiality obligations extending beyond the life of the patient [4].

An effective health information system integrates data, standardized indicators, established procedures, technological infrastructure, and human resources for clinical decision support and service improvement. According to Tasya *et al*. (2023), such integration creates harmony in operations and serves as a reference framework for improving health services, especially in medical record management [5]. The law in Indonesia requires reporting on mortality within 30 days after death occurs with responsibility placed on neighborhood authorities who must notify local implementing agencies about it [6]. Hospitals use this data on mortality to calculate their indices of mortality by types of deaths, causes of death classified into periodic reporting obligations monthly quarterly semi-annually annually. Manual reporting on mortality using spreadsheets exposes hospitals to three critical vulnerabilities: Data duplication happens very often because there is no automated validation mechanism; staff has to cross-check entries manually which is always prone to human oversight. Document integrity is not guaranteed because files can be corrupted or deleted by mistake or not properly saved. Most importantly, periodic aggregation takes a lot of time and effort manually; quarterly reports take 1-2 working days just for reformatting and cross-unit coordination [7][8].

There are still problems with the accuracy of certificates and compliance with procedures in reporting mortality in Indonesia. More than 80% of resident physicians at Sanglah General Hospital do not know how to fill out the death certificate according to ICD-10 [9]. These are systemic gaps in training medical personnel, quality assurance of certificates, and consistency in reporting that need integrated electronic solutions with continuous professional development. We chose Extreme Programming (XP) methodology for system development because it is structured but flexible for software development. XP, as an agile framework, puts more emphasis on customer responsiveness by involving users directly from the beginning stages of development so that requirements can grow organically throughout the project life cycle. Premana Putra *et al*. (2023) also stated that XP iterative approach allows quick changes to requirements while keeping development momentum [10]. This methodology uses simplified development models that support easy adjustments, increase flexibility, and intensify communication between developers and end users [11].

The Laravel Framework was selected as the development platform because it has efficient database operations, especially in optimizing queries and managing table relationships. Terlia and Firdonsyah (2024) said that Laravel architecture reduces initial development and maintenance costs while improving user experience through expressive and intuitive syntax [12][13]. Preliminary observations at Hospital X found very inefficient management of mortality data where medical record officers recorded patient deaths manually into spreadsheets, exported them to Excel format, filtered records by certain time ranges, reformatted them according to hospital standards, then submitted reports to the hospital director. This multi-step process takes too long, opens up opportunities for errors, and delays important reports. Therefore this study intends to develop a web-based mortality reporting information system that can process data quickly, accurately, and timely so as to support hospitals in managing integrated patient death reports within healthcare digitalization.

2. Related Work

Web-based death reporting systems have been the subject of numerous research studies, particularly as a means of enhancing the accuracy and efficiency of death data reporting at the village, hospital, and government agency levels. Several earlier research efforts have developed similar systems; however, they take different stances on the use of frameworks and software development techniques. Wulandari (2022) carried out a relevant study to expedite death reporting at the village level by developing a web-based solution [14]. The results show that digital methods can circumvent the recording errors and delays that older methods typically produce. Although the development techniques are not specifically explained, the study's goals align with the primary focus on information accessibility and efficiency.

Previous studies have shown that slower death reporting is caused by manual processing of death data, which includes patient analysis and searching. Apriliani *et al*. (2024) designed a system using Microsoft Visual Studio 2010 and Microsoft Access, with report results in the form of PDF files [15]. The design helps medical recorders reduce the time needed to fill, search, and record death report data. However, the software used cannot support the current advancement of web-based digital health technology because it can only be used on Windows operating systems, leading to limited access, integration, and collaboration across units within the hospital. As a solution to these problems, we chose to use the Laravel Framework to design a death reporting system. Laravel is a web-based framework that supports modern application development with MVC (Model-View-Controller) architecture. The use of Laravel allows the system to be accessed cross-platform through internet-connected devices, making it more flexible and not limited to one operating system. Laravel provides built-in security features such as authentication, encryption, and CSRF protection that are essential in managing sensitive medical data. The framework also has a wide community and complete documentation, making it easier to develop and maintain the system in the long run. Laravel supports integration with modern technologies such as APIs and health interoperability standards, allowing the death reporting system to connect with other modules in the hospital information system. With these advantages, the use of Laravel is considered more relevant to address the current challenges of healthcare digitization, especially in improving the efficiency, security, and integration capabilities of web-based death reporting systems [16][17][18].

In recent years, the Extreme Programming (XP) approach has been increasingly applied in software development in the healthcare sector, especially in hospitals and clinics that require speed of system development, data accuracy, and active user participation. XP is considered suitable because it is collaborative, iterative, and adaptive to evolving needs. Extreme Programming emphasizes technical aspects such as pair programming, continuous integration, test-driven development (TDD), and short iterations that allow quick feedback from users. Research shows that the application of XP in a hospital mortality data processing system can accelerate continuous improvement and produce valid, consistent, and faster mortality reports even though data comes from various units. XP also supports more intensive communication between developers and users so that adjustments to report formats and data sources can be made immediately. Another study highlighted the significance of digitizing death records, which were previously based on manual spreadsheets, and then replaced with an XP-based information system for issuing inpatient death certificates. The change made the mortality recording and reporting process more systematic, efficient, and accurate [3][19][7][20]. Development of an electronic system for death certificates using an Agile approach increased the speed and accuracy of document generation, also designed as a death monitoring application for health centers and hospitals with an interface that was rated favorably through the UX test [2][21].

The implementation of Extreme Programming (XP) on the general mortality reporting module at Hospital X showed significant quantitative impact through the application of the core practices of pair programming, test-driven development (TDD), and continuous integration (CI). Pair programming practices were shown to reduce syntax errors by 30% and accelerate collaborative problem solving. Chung *et al*. (2022) found that most death certificates in South Korea (around 95%) contained at least one error with the majority being "major errors," potentially distorting mortality statistics and requiring education and feedback-based interventions to improve reporting accuracy [23]. Evidence of country-scale implementation is also seen in Norway, where mandatory online death certificates automatically coded with ICD-10 through Iris improved the quality of death registration and impacted cause-of-death statistics [22][24].

An integrated approach that combines the Extreme Programming method with the Laravel framework allows for examining the integration between iterative software development methodologies and advanced web development technologies, providing advantages in terms of development speed, software quality, and ease of system maintenance. The research has the potential to make scientific contributions in two aspects. First, the study fills a gap in the literature regarding the application of XP in the development of web-based death reporting systems using Laravel. Second, the study provides empirical evidence of the effectiveness of the XP methodology in improving the quality, speed, and acceptance of systems in hospital environments, which in turn is expected to strengthen the national health information system through fast, accurate, and integrated death reporting. These studies demonstrate the significance of digitization in death reporting. However, literature specifically discussing the application of XP and Laravel in the development of death reporting systems remains scarce. Therefore, the study aims to fill the gap with a more adaptive and modern approach, as well as open up research opportunities to examine how the combination of these two methods can improve software quality, accelerate development time, and increase user adoption in health facilities.

3. Research Method

3.1 Data Collection Methods

Qualitative and quantitative descriptive research methodologies were used in the study to gain a deeper understanding of the process of developing a death reporting system at Hospital X, with an emphasis on interpretation, explanation, and contextual knowledge [25]. In the research and development team, there are several key roles: developers who are tasked with building and customizing system modules, medical record representatives who provide requirements, feedback, and validation from the user side, and testers who verify the functionality and quality of the software. The study also documents the implementation of relevant Extreme Programming (XP) core practices. Using qualitative descriptive research methods, the study successfully explored deeper insights into death reporting at Hospital X. Researchers sought to understand the perspectives of users and related parties. The researchers used observation, interviews, and literature review as data collection methods. Meanwhile, quantitative research methods used measurements of the average time taken to compile reports before and after the system was implemented, the level of accuracy of the data produced, and the number of recording errors that occurred.

3.1.1 Observation

Observations were made by observing and recording problems that occurred at Hospital X by looking at and understanding how the death data processing process took place in the field. During activities at the hospital, the author studied the death data management process in the medical records unit, starting from the receipt and verification of death certificates from various care units. Medical recorders ensured the completeness of documents such as patient identity, diagnosis, and time and place of death. Next, the cause of death was coded using ICD-10, and the data was entered into a spreadsheet that could only be accessed by authorized medical record officers. The collected data is then processed manually to generate mortality reports. The process still relies on downloading Excel files and filtering data based on quarterly periods (every 3 months) and cause of death categories, which requires more time before the reports can be submitted to the head of the medical records unit and the hospital director. The situation indicates delays in processing and potential data inaccuracies. Therefore, a solution is needed in the form of an integrated digital information system that can help medical record officers manage data more efficiently, accurately, and in a timely manner.

3.1.2 Interview

Researchers conducted interviews with medical record officers to gain an in-depth understanding of the death data processing flow, from input to output. During a week spent in the medical records unit, semi-structured interviews were conducted with officers responsible for mortality data management. In addition to interviews, researchers also observed manual workflows in the field and analyzed documents such as death certificates and mortality reports used by hospitals. The combination of methods provided a complete picture of the mechanisms of recording, verifying, and reporting deaths. Medical recorders explained that the coding process uses ICD-10 for causes of death and ICD-9 CM for procedures, with an emphasis on code accuracy because the data forms the basis of official hospital reports. Regarding the ultimate goal, mortality reports are used to evaluate service quality, identify the main causes of death, and fulfill routine reporting obligations to the Health Office. From these interviews and analysis, the researchers understood that accurate mortality reporting requires not only mastery of medical coding standards but also coordination across service units. The procedure is not merely administrative but a crucial part of efforts to improve the quality of health services.

3.1.3 Literature Review

Researchers collected data by reading and recording literature from various journal references of previous research results, then comparing information, grouping the data obtained, and drawing conclusions relevant to the research objective: the application of the Extreme Programming methodology in the development of a web-based death reporting system using the Laravel framework. From the analysis, it is hoped that solutions and recommendations will emerge to improve the effectiveness and efficiency of processing patient mortality data. The process also involved searching for reference sources from textbooks on medical records, scientific journals, Ministry of Health guidelines, and international standards such as ICD-10 published by the WHO. Literature searches were conducted online through journal portals and scientific repositories. Each piece of literature found was read and selected based on its relevance to the topic, such as definitions of mortality, procedures for coding causes of death, reporting processes in healthcare facilities, and the benefits of mortality data for evaluating service quality. The sources used were then recorded and grouped according to the discussion. The review results provided a thorough understanding of the principles, procedures, and standards for mortality reporting, which will serve as a reference for the preparation of the observation report. The literature review also helped identify knowledge gaps that could be further examined through field observations and interviews, and served as a theoretical foundation and reference for understanding and analyzing the mortality reporting process in hospitals.

3.2 System Development Methods

The analytical system's development process is founded on the Extreme Programming (XP) methodology, which seeks to increase system development's efficiency, speed, and organization. Kent Beck, Ron Jeffries, and Ward Cunningham developed Extreme Programming (XP), a software development technique recognized for its simple interface and reputation as an agile methodology. XP seeks to create specialized teams with a small to moderate number of individuals in order to do away with the necessity for large teams. Because Extreme Programming (XP) is an object-oriented software development method and its objective is software with requirements that are unclear at first and likely to change quickly, it was selected as the system development method for the study [10]. The following are the steps of the XP method:

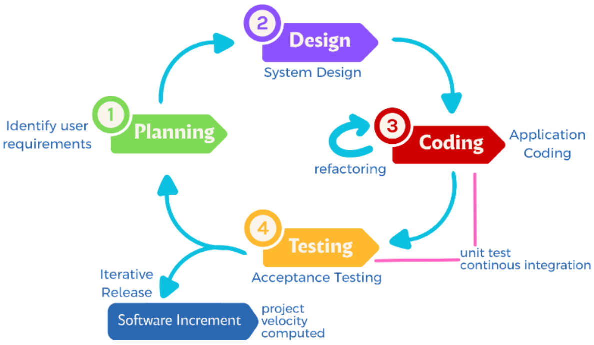

Figure 1. Extreme Programming (XP) Method

3.2.1 Planning

At the stage, observations and interviews were conducted, followed by a literature review referring to scientific works. After that, the author collected mortality data, then identified related issues related to the research, and determined the time required for system design and development. The design was carried out by analyzing the architecture and technology to be used in the research, analyzing the processes and flow of the system to be implemented, specifying user and data requirements as well as database specifications, and gaining an in-depth understanding of the Laravel framework, starting from describing the results, features on the website, the capacity of the website being developed, and the sequence of the website's progress [26][27].

3.2.2 Design

The design stage is the second stage of qualitative techniques, which is carried out after the planning stage is complete to model the system, including Mortality Flowmap, Mortality Context Diagram, Data Flow Diagram (DFD) Level 0, and Entity Relation Diagram (ERD) which can explain the flow of data processed to produce the desired information. The design phase also involves designing the architecture to be used, the architecture of the tools or technology to be used, and the data flow in the technology to be used. The existing database uses MySQL [30][31].

3.2.3 Coding

The phase uses the Laravel framework to build applications because it supports MVC ORM (Eloquent), middleware for security, and integration with Blade Template and Laravel Excel / DOMPDF for reports. In the programming stage of Extreme Programming (XP) modeling, the main focus is on writing code or implementing previously created designs. XP encourages pair programming, a method in which two programmers work together to develop existing source code, improving code quality and the team's understanding of the system [32][33].

3.2.4 Testing

Testing coding is carried out directly by medical recorders and nurses to determine that the system components that have been created are functioning properly. At the stage, the author conducts Black-Box testing, which focuses more on the functional requirements of the software, to evaluate the planned and developed system. Testers can define several input conditions under which testing will be performed and the expected results of the software. If there are bugs in the system, they will be fixed at the programming stage [34][35].

4. Result and Discussion

4.1 Results

The table shows that the manual process using spreadsheets takes longer at every stage, from patient input to report recapitulation. The web-based system is able to significantly reduce time, proving that digitizing mortality reporting increases efficiency while reducing the risk of human error

Table 1. Comparison of Mortality Data Entry Speeds

No.	Action	Manual (Spreadsheet)	Web-Based System (Laravel)	Explanation
1.	Input data on deceased patients (1 patient)	± 7–10 minutes	± 2–3 minutes	In the manual system, medical recorders need to open medical records, adjust death information by filling in columns on a spreadsheet, search for ICD-10 codes, and save them. In the web-based system, they only need to fill in a structured form, enter death information, search for ICD-10 codes, and save them.
2.	Validating duplicate data	± 10 minutes/patient	Automatic (< 1 second)	Manuals require checking one by one; web systems perform automatic validation.
3.	Quarterly recap	± 2–3 hours	± 10–20 seconds	The manual requires combining data from various sheets/units; the web system provides automatic reports according to the filtered time range.
4.	Prepare quarterly reports/each doctor	± 1–2 working days	± 30–50 seconds	The manual requires reformatting in accordance with hospital regulations and needs confirmation from the relevant unit; the web system generates reports according to the filters that have been set.

The following are the results of patient death reports generated using the information system, which inputs 35 data points from 3 rooms and 4 doctors in a year, then displays them in quarters I-IV:

PEMERINTAH KOTA BANDUNG
RUMAH SAKIT X BANDUNG
Jl. Rumah Sakit No.20, Pakemitan, Kec. Cinambo, Kota Bandung, Jawa Barat 45474

Laporan Data Kematian Di Rumah Sakit X Bandung
Dari Bulan January - March 2024

No	No RM	No Registrasi	Nama Pasien	Umur	Jenis Kelamin	Tanggal Kematian	Ruang Rawat	DPJP	Diagnosa	Kode ICD
1	241535	0001	AJANG SUMARNO	25	L	10-01-2024	ICU	dr. Anggi Jane, Sp.Bs.	Stroke Infark	I63.5
2	271535	0002	YOSI ROSMIATIN	24	P	15-01-2024	ICU	dr. Anggi Jane, Sp.Bs.	CKD	N18
3	391535	0003	ANITA ERNIWATI	46	P	20-01-2024	TULIP	dr. Wita Noor, Sp.PD	Chronic kidney disease, stage 5	N18.5
4	263005	0004	SITI MAEMUNAH	65	P	04-02-2024	ANGGREK	dr. Andy Jaelani, Sp.B	Cardiac arrest	I46
5	843312	0005	MARNI PURWANINGSIH	54	P	10-02-2024	ANGGREK	dr. Dina Rostiati, Sp.JP	Atherosclerotic heart disease	I25.1
6	942533	0006	ADEEVA AZZAHRA	25	P	19-02-2024	ICU	dr. Anggi Jane, Sp.Bs.	Cardiac arrest	I46
7	401535	0007	YAYAN SURYA	42	L	05-03-2024	TULIP	dr. Wita Noor, Sp.PD	Cardiac arrest	I46
8	288927	0008	NASEP FAJRI IMANI	50	L	17-03-2024	TULIP	dr. Dina Rostiati, Sp.JP	Hypertensive heart disease with (congestive) heart failure	I11.0

Figure 2. Quarterly I Data Results

PEMERINTAH KOTA BANDUNG
RUMAH SAKIT X BANDUNG
Jl. Rumah Sakit No.20, Pakemitan, Kec. Cinambo, Kota Bandung, Jawa Barat 45474

Laporan Data Kematian Di Rumah Sakit X Bandung
Dari Bulan April - June 2024

No	No RM	No Registrasi	Nama Pasien	Umur	Jenis Kelamin	Tanggal Kematian	Ruang Rawat	DPJP	Diagnosa	Kode ICD
1	748632	0009	TATA CALISTA	27	P	07-04-2024	ANGGREK	dr. Anggi Jane, Sp.Bs.	Cerebral edema	G93.6
2	451535	0010	IRAWATI BINTI JUNAEDIN	48	L	17-04-2024	ICU	dr. Wita Noor, Sp.PD	Type 2 diabetes mellitus with kidney complications	E11.2
3	521112	0011	HERMAWAN BIN JAMAL	46	L	07-05-2024	TULIP	dr. Wita Noor, Sp.PD	Alcoholic cirrhosis of liver	K70.3
4	431535	0012	DEWI KURAESIH	44	P	16-05-2024	ANGGREK	dr. Wita Noor, Sp.PD	Acute hepatitis C	B17.1
5	700926	0013	RISWANA BIN TONO	25	L	20-05-2024	ICU	dr. Anggi Jane, Sp.Bs.	PPOK	J44.1
6	301535	0014	IDANG TIMANI	47	L	03-06-2024	TULIP	dr. Wita Noor, Sp.PD	Chronic kidney disease, stage 5	N18.5
7	361535	0015	NABILA FITRIANA	26	P	11-06-2024	ICU	dr. Anggi Jane, Sp.Bs.	Cerebral edema	G93.6

Figure 3. Quarterly II Data Results

PEMERINTAH KOTA BANDUNG
RUMAH SAKIT X BANDUNG
Jl. Rumah Sakit No.20, Pakemitan, Kec. Cinambo, Kota Bandung, Jawa Barat 45474

Laporan Data Kematian Di Rumah Sakit X Bandung
Dari Bulan July - September 2024

No	No RM	No Registrasi	Nama Pasien	Umur	Jenis Kelamin	Tanggal Kematian	Ruang Rawat	DPJP	Diagnosa	Kode ICD
1	131535	0016	SAM KURNIAWAN	33	L	07-07-2024	ANGGREK	dr. Anggi Jane, Sp.Bs.	Meningitis	G03.9
2	551535	0017	SAKTI GUNA MADJI	42	L	15-07-2024	ANGGREK	dr. Andy Jaelani, Sp.B	Liver cell carcinoma	C22.0
3	441535	0018	DWI SEPTIANI	43	P	21-07-2024	TULIP	dr. Wita Noor, Sp.PD	Essential (primary) hypertension	I10
4	277910	0019	TIFFANY DEWI	47	P	02-08-2024	ANGGREK	dr. Wita Noor, Sp.PD	Acute hepatitis A without hepatic coma	B15.9
5	781435	0020	UJANG SOLEH	56	L	19-08-2024	ICU	dr. Dina Rostiati, Sp.JP	Congestive heart failure	I50.0
6	560835	0021	SUMIYATI BINTI GANI	58	P	03-09-2024	ICU	dr. Anggi Jane, Sp.Bs.	Anoxic brain damage	G93.1
7	481335	0022	PUTRI GUSVIANNA	42	P	12-09-2024	ANGGREK	dr. Wita Noor, Sp.PD	Malignant neoplasm of main bronchus	C34.0
8	261535	0023	ALISA RIMA	36	P	06-09-2024	ANGGREK	dr. Anggi Jane, Sp.Bs.	Atherosclerosis	I70.9
9	502013	0024	ZARA ALESHA	41	P	21-09-2024	TULIP	dr. Wita Noor, Sp.PD	Tuberculosis of lung, confirmed by sputum microscopy with or without culture	A15.0

Figure 4. Quarterly III Data Results

PEMERINTAH KOTA BANDUNG
RUMAH SAKIT X BANDUNG
Jl. Rumah Sakit No.20, Pakemitan, Kec. Cinambo, Kota Bandung, Jawa Barat 45474

Laporan Data Kematian Di Rumah Sakit X Bandung
Dari Bulan October - December 2024

No	No RM	No Registrasi	Nama Pasien	Umur	Jenis Kelamin	Tanggal Kematian	Ruang Rawat	DPJP	Diagnosa	Kode ICD
1	886811	0025	ALIT BINTI ENANG	32	L	04-10-2024	ICU	dr. Andy Jaelani, Sp.B	Malignant neoplasm of rectum	C20
2	571630	0026	DADAN ABDUROHMAN	59	L	10-10-2024	ICU	dr. Dina Rostiati, Sp.JP	Cardiac arrest	I46
3	841734	0027	RUSMINAH BINTI ODENG	42	P	17-10-2024	ICU	dr. Andy Jaelani, Sp.B	Gastroenteritis	A09
4	605405	0028	EUIS YULIYATI	32	P	23-10-2024	ICU	dr. Dina Rostiati, Sp.JP	Hypertension	I10
5	667412	0029	FAIZAN DWI PUTRA	46	L	02-11-2024	TULIP	dr. Wita Noor, Sp.PD	Cardiac arrest	I46
6	598733	0030	DIAN SYAHRANI	24	P	12-11-2024	ICU	dr. Anggi Jane, Sp.Bs.	Traumatic subdural hemorrhage	S06.5
7	581535	0031	RAMADHANISA BINTI AJI	54	P	17-11-2024	ANGGREK	dr. Andy Jaelani, Sp.B	Acute kidney failure	N17
8	396604	0032	KHOLPI MAHARANI MAHMI	46	P	03-12-2024	TULIP	dr. Dina Rostiati, Sp.JP	Chronic kidney disease, stage 5	N18.5
9	874221	0033	MAHENDRA THORIQ	29	L	07-12-2024	ANGGREK	dr. Andy Jaelani, Sp.B	Hepatic failure	K72
10	421535	0034	KIKI PRATAMA	43	L	11-12-2024	TULIP	dr. Wita Noor, Sp.PD	Malignant neoplasm of breast of unspecified site	C50.9
11	351535	0035	OON ROHANI	44	P	22-12-2024	ANGGREK	dr. Anggi Jane, Sp.Bs.	Cardiac arrest	I46

Figure 5. Quarterly IV Data Results

Figures 2-5 show the results of patient mortality reports printed in PDF format based on data filtered from the first to fourth quarters for submission to the head of medical records and the hospital director.

PEMERINTAH KOTA BANDUNG
RUMAH SAKIT X BANDUNG
Jl. Rumah Sakit No.20, Pakemitan, Kec. Cinambo, Kota Bandung, Jawa Barat 45474

Laporan Data Kematian Di Rumah Sakit X Bandung DPJP: **dr. Wita Noor, Sp.PD**

No	No RM	No Registrasi	Nama Pasien	Umur	Jenis Kelamin	Tanggal Kematian	Ruang Rawat	DPJP	Diagnosa	Kode ICD
1	391535	0003	ANITA ERNIWATI	46	P	20-01-2024	TULIP	dr. Wita Noor, Sp.PD	Chronic kidney disease, stage 5	N18.5
2	401535	0007	YAYAN SURYA	42	L	05-03-2024	TULIP	dr. Wita Noor, Sp.PD	Cardiac arrest	I46
3	451535	0010	IRAWATI BINTI JUNAEDIN	48	L	17-04-2024	ICU	dr. Wita Noor, Sp.PD	Type 2 diabetes mellitus with kidney complications	E11.2
4	521112	0011	HERMAWAN BIN JAMAL	46	L	07-05-2024	TULIP	dr. Wita Noor, Sp.PD	Alcoholic cirrhosis of liver	K70.3
5	431535	0012	DEWI KURAESIH	44	P	16-05-2024	ANGGREK	dr. Wita Noor, Sp.PD	Acute hepatitis C	B17.1
6	301535	0014	IDANG TIMANI	47	L	03-06-2024	TULIP	dr. Wita Noor, Sp.PD	Chronic kidney disease, stage 5	N18.5
7	441535	0018	DWI SEPTIANI	43	P	21-07-2024	TULIP	dr. Wita Noor, Sp.PD	Essential (primary) hypertension	I10
8	277910	0019	TIFFANY DEWI	47	P	02-08-2024	ANGGREK	dr. Wita Noor, Sp.PD	Acute hepatitis A without hepatic coma	B15.9
9	481335	0022	PUTRI GUSVIANNA	42	P	12-09-2024	ANGGREK	dr. Wita Noor, Sp.PD	Malignant neoplasm of main bronchus	C34.0
10	502013	0024	ZARA ALESHA	41	P	21-09-2024	TULIP	dr. Wita Noor, Sp.PD	Tuberculosis of lung, confirmed by sputum microscopy with or without culture	A15.0
11	667412	0029	FAIZAN DWI PUTRA	46	L	02-11-2024	TULIP	dr. Wita Noor, Sp.PD	Cardiac arrest	I46
12	421535	0034	KIKI PRATAMA	43	L	11-12-2024	TULIP	dr. Wita Noor, Sp.PD	Malignant neoplasm of breast of unspecified site	C50.9

Figure 6. Data Results by Doctor

Figure 6 shows the results of the patient death report printed in PDF format according to the data of the patient's attending physician to be reported to the head of medical records and the hospital director.

4.2 Discussion

Based on the stages of the Extreme Programming method, the following was carried out:

4.2.1 Planning

The planning stage began with direct observation of the death data processing workflow and interviews with relevant medical recorders to obtain an overview of the problems occurring in the field. The author also conducted a literature review of scientific works to understand best practices in mortality reporting and ICD coding standards. Based on the data collected, system requirements were identified, including entities needed for the database, such as: patient identity, time of death, cause of death, ICD classification, and users (medical recorders and nurses). Role-based access control is established at the stage: nurses can only enter and edit patient data, while medical record officers have the right to review, filter, and print reports. The estimated duration of system development is determined based on the complexity of requirements and the estimated number of development iterations. During the planning stage, the author involved 2–4 users (nurses and medical recorders) to provide initial input, thereby maintaining transparency and accountability in the requirements gathering process [26].

4.2.2 Design

At the stage, an overview of the patient death reporting system design is presented. The system design uses Flowmaps, Context Diagrams, Data Flow Diagrams, and Entity Relationship Diagrams (ERD).

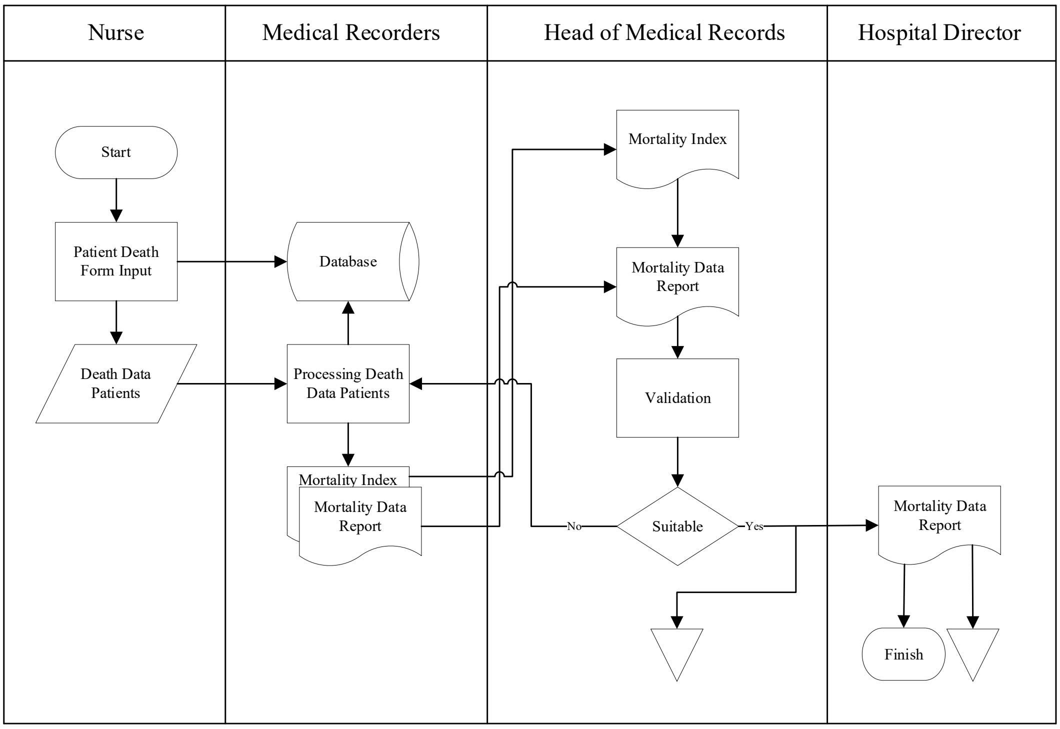

Figure 7. Mortality Flowmap

A flowmap or flowchart is a graphical representation of various methods, procedures, or processes [28]. The Death Mortality explains how the system starts with nurses entering Patient Death Forms, then medical recorders receive and process the data to generate databases, death indexes, and death reports.

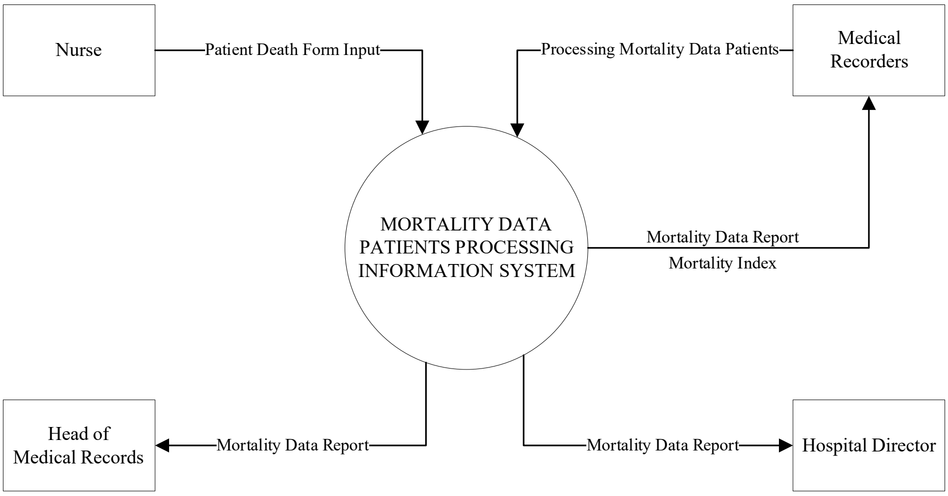

Figure 8. Context Diagram

A context diagram is a high-level data diagram that describes the system as a whole. The diagram contains one main process and shows how the system interacts with external entities [29].

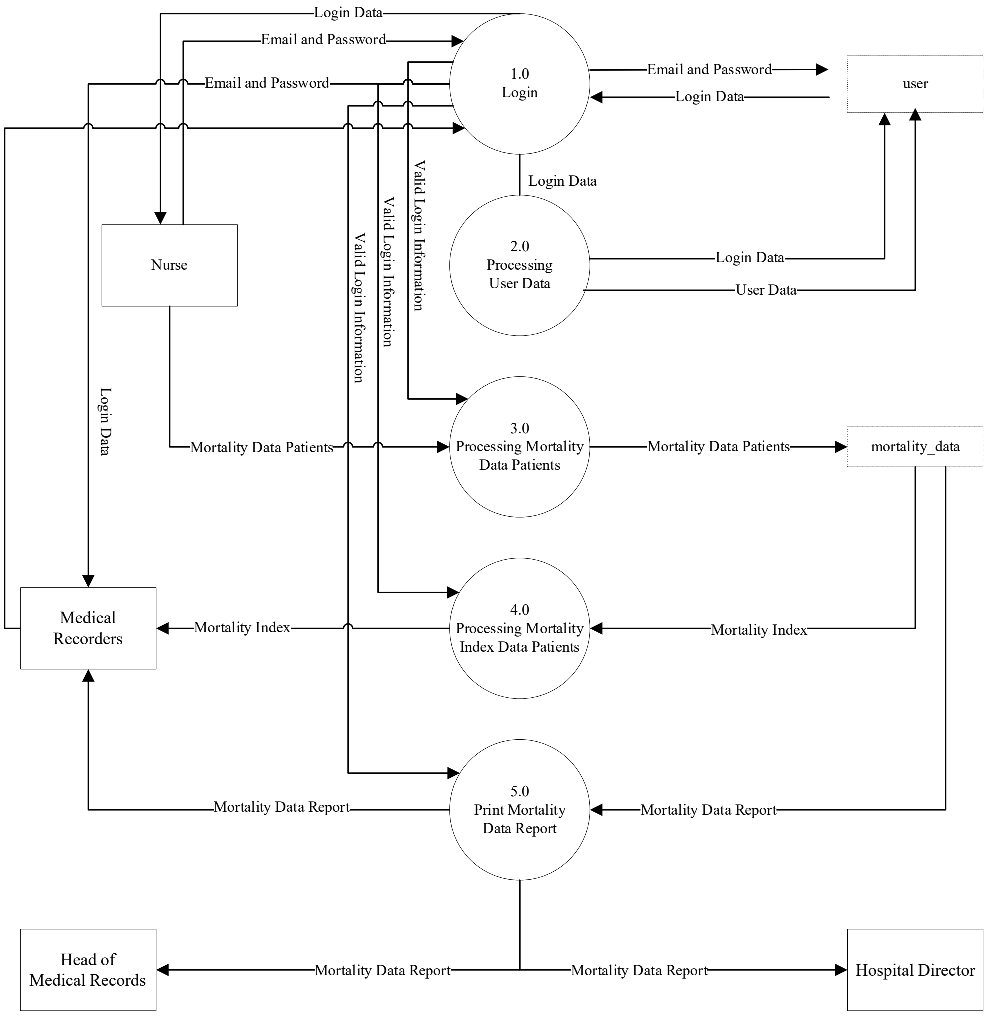

Figure 9. Data Flow Diagram Level 0

The logic of data is displayed by a Data Flow Diagram (DFD), which shows the origins of the data, its destination when exiting the system, the database of the system, the operations that produce it, and the connections between the data held by the processes that utilize it.

Table 2. DFD Explanation

No.	Data Flow Diagram	Explanation
1.0	Login	Login by Nurse and Medical Recorders as users to access the system
2.0	Manage User Data	After Nurse and Medical Recorders successfully log in as users, their data is automatically stored in the system
3.0	Manage Patient Mortality Data	Nurse inputs the patient's mortality data, which is then managed by the Medical Recorders
4.0	Manage Mortality Index	Medical Recorders manage mortality data, which generates mortality index data
5.0	Print Mortality Reports	Medical Recorders printing mortality reports after they have been managed and filtered as needed for submission to the Medical Records Manager and Hospital Director

Table 2 explains the DFD of how existing entities perform activities in the system, from input to output.

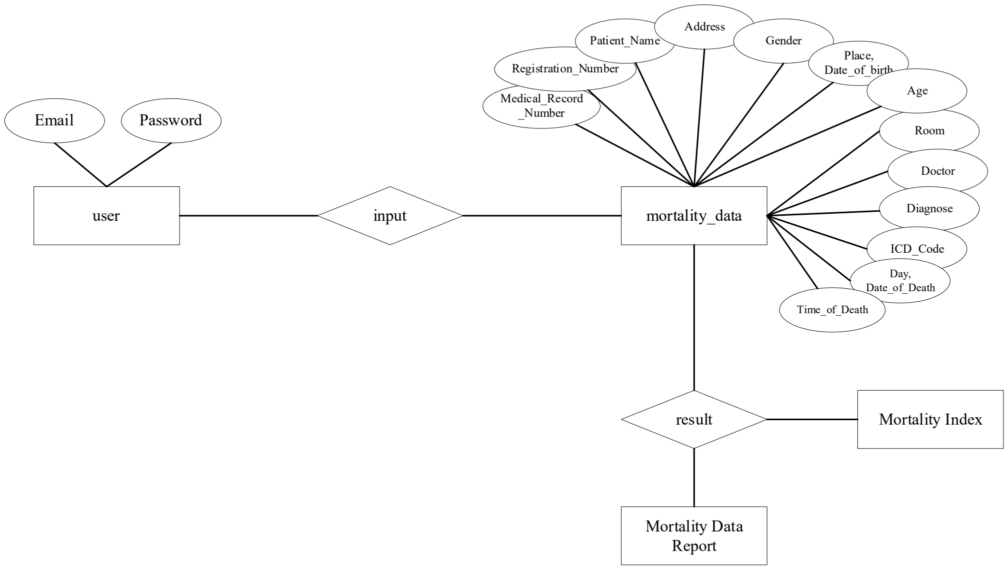

Figure 10. Entity Relationship Diagram

The entity relation diagram of the web-based death reporting system illustrates how users interact with the system to enter death data, as well as how the system produces output in the form of reports and death indexes.

4.2.3 Coding

The programming stage involves implementing the features of the mortality reporting information system based on the design that has been created. The system is developed iteratively using pair programming practices, in accordance with the principles of Extreme Programming (XP), which emphasizes team collaboration and continuous improvement. The programming stage is carried out iteratively using pair programming practices in accordance with XP principles, enabling continuous improvement and collaboration between developers. The stage realizes the design created in the previous stage, including the implementation of a database that covers important entities. The program code is arranged to support relations between entities, data integrity, and to facilitate the retrieval of information for quarterly reports and filters for each doctor. In addition, role-based access control is implemented in the code: nurses can only enter, edit, and view patient data, while medical record officers have the right to review, filter, and print reports. Development is carried out in a modular manner so that any feature additions or access rights changes can be applied without disrupting other parts of the system. During the stage, the author continues to coordinate with end users to ensure that the implementation meets the needs of the hospital [32]. The stages of development using Laravel in the mortality reporting information system include several important steps to build a stable, secure, and integrated application. Here are the general stages:

1) Project Setup: The first step is to create a new Laravel project using the Laravel CLI command. The project will serve as the framework for developing a web-based mortality reporting information system application.
2) Database Design: Once the project is created, the next step is to design the database structure needed to store information such as patient identity, time of death, cause of death based on ICD classification, treatment room, doctor, and users with different roles (nurses and medical recorders). This involves using migration to create tables and appropriate inter-table relationships.
3) Model Development: Laravel models are used to manage data access in the database. Each model will represent entities in the system such as patient identity, death data, or death reports. These models will be used to perform CRUD (create, read, update, delete) operations on the data.
4) Controller Development: Laravel controllers are responsible for managing the flow of business logic in the application. These controllers handle HTTP requests from users, process input, and interact with models to retrieve or update data. Nurses use controllers to input patient death data, while medical recorders use controllers to filter and generate reports.
5) User Interface (UI/UX) Development: The user interface in the mortality reporting information system needs to be designed using Laravel templating technology such as Blade. This involves creating displays that will be shown to users, such as login pages, dashboards, patient death data input, death indices, death data, and death data reports.
6) Functionality Implementation: Once the application base has been built, the next step is to implement key functionalities such as user authentication, mortality data management, and mortality reporting.

7) Testing: The next important step is to test the web-based mortality reporting information system application using Laravel features such as PHPUnit or Laravel Dusk. The testing will ensure that the application functions properly and meets user needs.
8) Security Settings: The final step is to set up application security by implementing features such as user authentication, CSRF protection, and input validation to protect sensitive data and prevent security attacks. By following these steps systematically, developers can build a reliable, efficient, and integrated mortality reporting information system that meets user needs using the Laravel framework.

The stage is carried out after the design stage is complete and the application has been approved by users. The following are the results of the programming stage applied to the mortality information system display that has been created.

Figure 11. Login Page

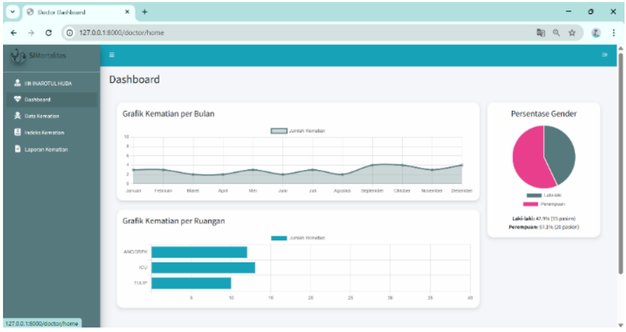

Figure 12. Dashboard Page Display

Officers access the system by logging in using their registered email address and password, ensuring that only verified users can log in. The system implements role-based access controls: nurses can only input patient and death data in their unit, while medical recorders can verify, manage, and print reports from all existing data. The mechanism improves security and accuracy, while also supporting audit trails. The development of the login module follows XP principles with small iterations and repeated validation, so that access rights can be adjusted to the needs of the hospital and these regulations (Figure 11). After successfully logging in, officers will be directed to the dashboard as the system's home screen. The dashboard displays key features according to role, including patient death input forms, patient death data, death indices, and death reports. With the design, officers can quickly access relevant modules, supporting process efficiency, data accuracy, and real-time monitoring of activities according to the authorization given. The display is available to every logged-in user and shows monthly death graphs, death graphs for each room, and death rates by gender between males and females (Figure 12).

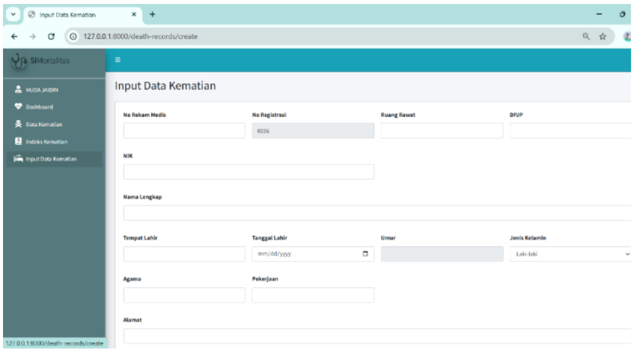

Figure 13. Patient Mortality Data Input Display

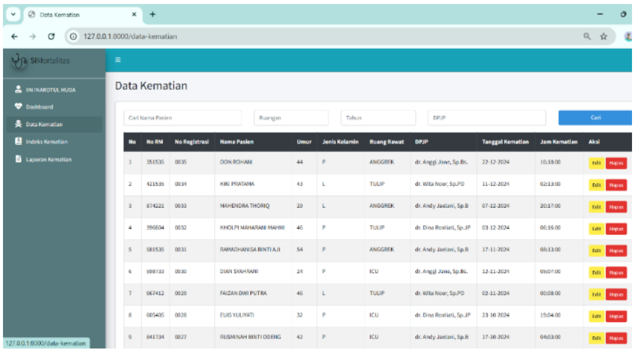

Figure 14. Patient Mortality Data Display

The form can be filled in by nurses to enter complete patient death data, including the date, day, and time of death. The diagnosis and ICD-10 code as the cause of death will be filled in by medical recorders when editing the data. The system validates each required field to ensure data completeness and accuracy, while facilitating the classification of causes of death according to international standards. With structured data entry, officers can reduce the risk of recording errors and facilitate the preparation of quarterly reports and individual doctor reports (Figure 13). Figure 14 displays the patient death data that has been entered, serving as a reference for reporting. The information displayed includes patient identity, date, day, and time of death, making it easier for officers to verify, monitor, and compile reports. The system also supports role-based access control, so that nurses can only view the data, while medical recorders can access all data for validation and reporting purposes (Figure 14).

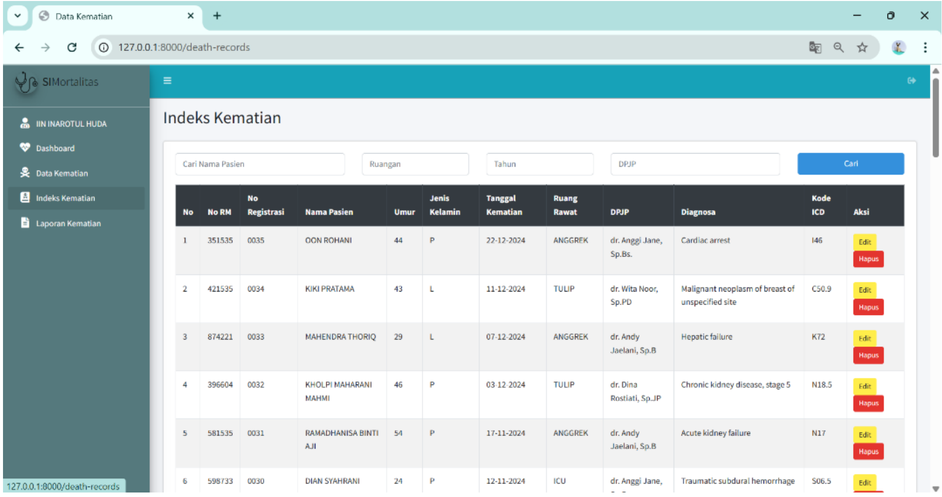

Figure 15. Mortality Index Display

The page displays the mortality index, which is calculated from validated patient death data that has been entered. The information displayed includes patient identity, time of death, diagnosis, and ICD-10 code. The module makes it easy for staff to monitor mortality trends, perform statistical analysis, and ensure reporting is in accordance with clinical and regulatory standards. The system also implements role-based access controls, so that nurses can only view data, while medical recorders can edit or delete data as needed for validation and reporting.

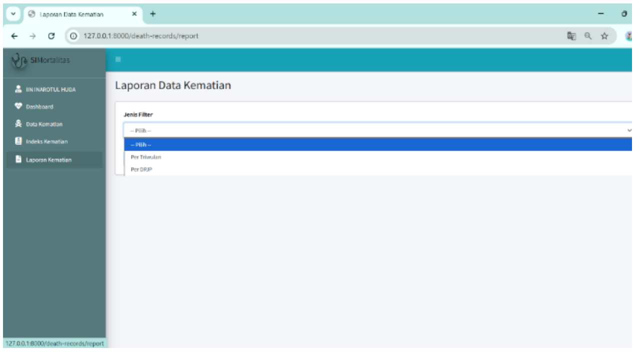

Figure 16. Data Filter Display for Mortality Reports

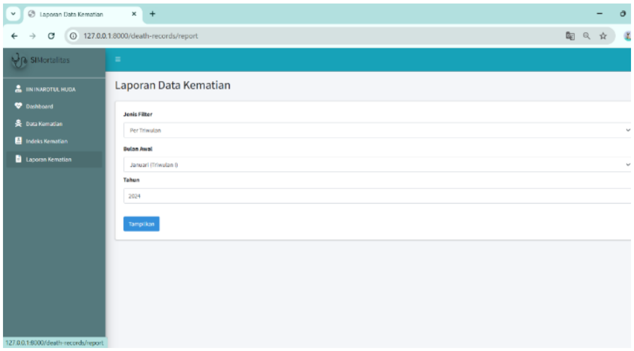

Figure 17. Quarterly Data Filter

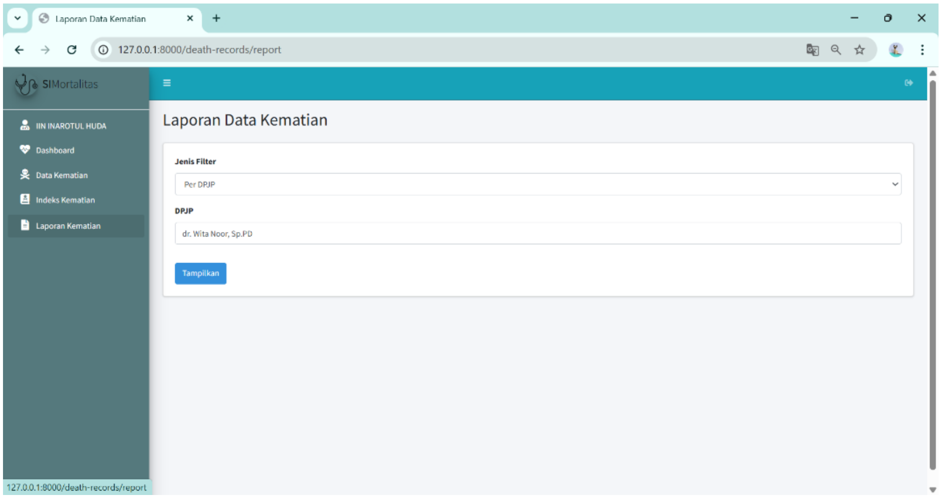

Figure 18. Doctor Data Filter

Figures 16, 17, and 18 show the death report filters generated after the data has been entered and validated. These reports can be filtered as needed, including by quarters I to IV and by the doctor responsible for the patient. The filtering feature makes it easier for medical recorders to analyze mortality trends, compile quarterly or per-doctor reports, and support factual and integrated data-driven decision making.

4.2.4 Testing

Testing was conducted using the Black-Box method, focusing on validating system inputs and outputs without regard to the internal structure of the code. Each test involved 2–4 end users, who provided feedback for system improvements. If bugs or inconsistencies with the design and user requirements were found, improvements were made immediately at the programming stage. XP emphasizes repeated testing and periodic integration, so that each module is continuously tested to ensure that the system is stable, secure, and fully operational before being implemented in a hospital environment [34][35]. Black box testing focuses on interfaces, inputs, and outputs, without examining the source code. In Laravel, the technique enables testing of modules such as death data input, dashboards, and reports from the user side. The XP methodology supports small iterations and rapid releases, so that functional issues can be detected early and the system can be tailored to the needs of the hospital. The aim is to evaluate system performance without focusing on implementation details, ensuring that the system functions properly and meets user requirements [36].

Table 3. Black Box System Testing

Testing Scenario	Expected Implementation	Test Report	Results
Log in by entering email and password	The system will display the user login page to log in	The system displays the user login page for logging in	Success
Entering data on the patient death form	The system will display the patient death registration form page for storage	The system displays the patient death registration form page and saves it	Success
Displaying patient mortality data	The system is capable of displaying patient mortality data that has been entered	Patient mortality data has been stored and can be displayed	Success
Displaying patient mortality index	The system is capable of displaying the mortality index of patients that have been entered	Patient mortality index data has been stored and can be displayed	Success
Printing patient death reports quarterly I-IV	The system is capable of displaying and printing patient mortality reports quarterly I-IV	Stored patient death data reports can be printed and downloaded in PDF format quarterly I-IV	Success
Printing patient death reports by doctor	The system is capable of displaying and printing reports on patient mortality data by doctor	The system is capable of displaying and printing reports on patient mortality data by doctor	Success
Printing patient mortality index	The system will be able to display and print patient mortality indices	The system is capable of displaying and printing patient mortality index	Success

5. Conclusion and Recommendations

The study demonstrates that the death data processing system at Hospital X, which is still done manually using spreadsheets, is not yet effective because it takes a long time and is prone to human error. The development of a web-based death reporting system using the Laravel framework and Extreme Programming (XP) methodology has proven to improve data processing effectiveness, reduce the risk of errors, provide real-time access, and support faster and more accurate decision making by hospital management. The system performs better than the manual method, reducing patient data input time to ± 2–3 minutes, automating data duplication validation, and reducing recap and report generation to just seconds. The application of Extreme Programming supports the achievement, particularly through pair programming, which speeds up the correction of critical defects. The implementation of Laravel and XP also ensures that the system remains flexible, responsive to user input, and capable of adapting to regulatory changes or future hospital needs. The system not only addresses current operational needs but also provides a foundation for sustainable hospital digital transformation, supporting efficiency, accuracy, and clinical data-driven decision-making.

The limitations of the research lie in the fact that it has not been tested in high-load scenarios. Going forward, development should focus on several key areas to enhance system capabilities and ensure long-term sustainability. First, implementing automatic ICD code integration through automated ICD-10 code suggestion features can reduce manual coding time and improve accuracy in cause-of-death classification. Second, developing granular audit logs with detailed audit trail mechanisms to track all user activities, data modifications, and system access for accountability and security purposes. Third, strengthening data security by implementing field-level encryption for sensitive patient information to comply with healthcare data protection regulations. Fourth, conducting cross-hospital testing through pilot implementations across multiple healthcare facilities to assess the system's scalability, interoperability, and adaptability to different hospital environments and workflows. Fifth, testing the system under high-load scenarios to ensure stability and responsiveness during peak usage periods, particularly when handling large volumes of mortality data. Sixth, establishing structured training programs for medical recorders and nurses to maximize system adoption and ensure proper utilization of all features. Finally, exploring opportunities to connect the mortality reporting system with national health databases to facilitate seamless data exchange and support public health surveillance efforts. These recommendations aim to address current limitations while positioning the system for broader implementation and integration within the national healthcare information infrastructure, ultimately contributing to improved mortality data quality and healthcare service delivery.

References

[1] Fadilah, R. A., Abdussalaam, F., & Yunengsih, Y. (2024). Perancangan sistem informasi konsolidasi integritas rekam medis rawat jalan berbasis web guna menunjang efektivitas pengklaiman BPJS di Rumah Sakit Hermina Arcamanik Bandung. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, *5*(2), 1921–1937. https://doi.org/10.35870/jimik.v5i2.818

[2] Nugraha, F. S., Syahidin, Y., & Suryani, A. I. (2024). Penerapan teknologi sistem informasi dalam proses pembuatan surat keterangan kematian berbasis elektronik menggunakan metode agile. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, *7*(2), 483–493. https://doi.org/10.32493/jtsi.v7i2.39134

[3] Ramadhan, F. D., Abdussalaam, F., & Sari, I. (2024). Perancangan sistem informasi pengolahan data kematian berbasis web guna menunjang efektivitas pelaporan mortalitas. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, *7*(3), 1102–1108. https://doi.org/10.32493/jtsi.v7i3.40348

[4] Permenkes No. 24. (2022). Peraturan Menteri Kesehatan Republik Indonesia Nomor 24 Tahun 2022. 2003–2005. https://doi.org/10.1136/bmj.p1915

[5] Tasya, A. H., Setiani, T., Syahidin, Y., & Yunengsih, Y. (2023). Tata kelola rekam medis berbasis elektronik dalam menunjang pelaporan operasi dengan menggunakan metode agile. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, *4*(3), 1265–1273. https://doi.org/10.35870/jimik.v4i3.377

[6] UU No. 24. (2013). *Undang-Undang Nomor 24 Tahun 2013 tentang Administrasi Kependudukan*. Retrieved from https://www.kemhan.go.id/ppid/wp-content/uploads/sites/2/2016/10/UU-24-TAHUN-2013.pdf

[7] Alfiani, D., Saputri, W., Syahidin, Y., & Setiatin, S. (2025). Design of mortality data processing information system to support accuracy of mortality reporting at Hermina Arcamanik Hospital. *International Journal of Information*, 0–8. https://doi.org/10.62951/ijies.v2i2.244

[8] Feijt, M. A., de Kort, Y. A. W., Westerink, J. H. D. M., Bierbooms, J. J. P. A., Bongers, I. M. B., & IJsselsteijn, W. A. (2021). Assessing professionals' adoption readiness for eMental health: Development and validation of the eMental health adoption readiness scale. *Journal of Medical Internet Research*, *23*(9). https://doi.org/10.2196/28518

[9] Adi, I. G., Jagadhita, L., Ayu, I. G., Ratnayanti, D., Sugiritama, I. W., Gusti, I., & Arijana, K. N. (2022). Knowledge and practice of filling out death certificates among resident doctors at Sanglah General Hospital. *Jurnal Medika Udayana*, *11*(2), 65–69. https://doi.org/10.24843.mu.2022.v11.i9.p02

[10] Premana Putra, I. G. B., Sudarma, M., & Manuaba, I. B. G. (2023). Penerapan metode extreme programming pada rancang bangun sistem analisis sentimen portal berita. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *10*(6), 1369–1378. https://doi.org/10.25126/jtiik.2023106904

[11] Najrifatussya'diah, S., Syahidin, Y., & Yunengsih, Y. (2024). Rancang bangun menu konfirmasi sebagai pendukung keputusan klaim jaminan kesehatan nasional menggunakan metode extreme programming. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, *7*(2), 643–652. https://doi.org/10.32493/jtsi.v7i2.39123

[12] Terlia, C., & Firdonsyah, A. (2024). Application of the Laravel framework in the development of a web-based information system for Biophysio Physiotherapy Clinic. *Antivirus: Jurnal Ilmiah Teknik Informatika*, *18*(2), 222–233. https://doi.org/10.35457/antivirus.v18i2.3953

[13] Putri, T. E., Apriyandi, D. T., & Ariani, S. (2024). Optimalisasi rekam medis kesehatan: Pengembangan sistem informasi SIREDI berbasis website dengan framework Laravel. *Jurnal Rekursif*, *12*(1). https://doi.org/10.33369/rekursif.v12i1.33720

[14] Desak Putu Sri Wulandari, I Ketut Gede Suhartana, & Ida Bagus Gede Dwidasmara. (2022). Sistem informasi pelaporan kematian pada tingkat dusun di Kabupaten Badung. *Jurnal Pengabdian Informatika*, *1*(1), 161–166. https://doi.org/10.24843/jupita.2022.v01.i01.p23

[15] Apriliani, I., Syahidin, Y., & Suryani, A. I. (2024). Desain formulir pelaporan kematian perinatal berbasis elektronik dengan menggunakan metode waterfall. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, *7*(4), 1451–1461. https://doi.org/10.32493/jtsi.v7i4.43838

[16] Mangca, D. C. (2023). Efficient healthcare data management: Developing a patient information system using Laravel framework. *International Journal of Innovative Science and Research Technology*, *8*(7). https://doi.org/10.5281/zenodo.8216645

[17] Putri, L. D., Setyawan, A. A., & Yunengsih, Y. (2024). Electronic medical record information system to support clinical management of web-based medical report resumes. *International Journal Software Engineering and Computer Science (IJSECS)*, *4*(3), 1233–1243. https://doi.org/10.35870/ijsecs.v4i3.3106

[18] Sekar, A., & Yuningsih, L. (2024). Pengembangan aplikasi dashboard pasien rawat inap pada Rumah Sakit XYZ menggunakan framework Laravel. *Jurnal Teknologi dan Sistem Informasi Bisnis*, *6*(3), 404–410. https://doi.org/10.47233/jteksis.v6i3.1379

[19] Sulastri, I. Y., Syahidin, Y., Gunawan, E., & Sukmawijaya, J. (2023). Rancang bangun sistem informasi surat keterangan kematian pasien rawat inap menggunakan metode extreme programming. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, *6*(2), 110–124. https://doi.org/10.32493/jtsi.v6i2.29691

[20] Hidayatuloh, F., Syahidin, Y., & Yunengsih, Y. (2024). Implementation of extreme programming in the design of hospital daily report recapitulation system. *Multidisciplinary International Journal of Research and Development*, *3*(5), 1–11. https://doi.org/www.mijrd.com/papers/v3/i5/mijrdv3i50001

[21] Muhriati, M. (2024). Perancangan aplikasi pelaporan surveilans mortality di Kabupaten Barru. *Sehatmas: Jurnal Ilmiah Kesehatan Masyarakat*, *3*(4), 701–712. https://doi.org/10.55123/sehatmas.v3i4.3815

[22] Martínez, G. de la C. (2024). Enhancing collaboration and code quality using pair programming. *International Journal of Software Engineering and Applications*, *15*(6), 1–15. https://doi.org/10.5121/ijsea.2024.15604

[23] Chung, S., Kim, S. H., Park, B. J., & Park, S. (2022). Factors associated with major errors on death certificates. *Healthcare*, *10*(4), 726. https://doi.org/10.3390/healthcare10040726

[24] Eng, H. M., Sveen, K. A., Fagerås, S. J., Strøm, M. S., Diep, L. M., Juliusson, P. B., & Alfsen, G. C. (2024). Introduction of electronic death notification in Norway—Impact on diabetes mortality registration. *PLOS ONE*, *19*(12), 1–11. https://doi.org/10.1371/journal.pone.0311106

[25] Mahdani, R., Yaumi, T., Syahidin, Y., & Yunengsih, Y. (2023). Tata kelola rekam medis berbasis elektronik dalam pembuatan laporan poliklinik pasien rawat jalan menggunakan metode agile. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, *4*(3), 1050–1060. https://doi.org/10.35870/jimik.v4i3.315

[26] Taufiq, G., & Handrianto, Y. (2022). Model extreme programming untuk rancang bangun sistem informasi penjadwalan kuliah. *Infortech*, *4*(2), 190–199. https://doi.org/10.31294/infortech.v4i2.14065

[27] Ardhana, V. Y. P. (2024). Penerapan metode extreme programming pada sistem informasi penjualan berbasis web. *Journal of Information Systems Management and Digital Business*, *1*(2), 227–235. https://doi.org/10.59407/jismdb.v1i2.414

[28] Azzahra, S. F., Syahidin, Y., Yunengsih, Y., & Law, C. (2025). Perancangan visum et repertum berbasis elektronik sebagai alat bukti tindak pidana dengan metode rapid application development. *DECODE: Jurnal Pendidikan Teknologi Informasi*, *5*(1), 121–134. https://doi.org/10.51454/decode.v5i1.1085

[29] Kustiani, R., Syahidin, Y., Yunengsih, Y., Information, H., Study, M., & Ganesha, P. P. (2024). Electronic medical record information system design to support the reporting of completeness of BPJS patient medical records. *JUTIF: Jurnal Teknik Informatika*, *5*(4), 11–19. https://doi.org/10.52436/1.jutif.2024.5.4.1976

[30] Fajar Mahardika, Sania Galuh Merani, & Akrim Teguh Suseno. (2024). Penerapan metode extreme programming pada perancangan UML sistem informasi penggajian karyawan. *Blend Sains Jurnal Teknik*, *2*(3), 204–217. https://doi.org/10.56211/blendsains.v2i3.313

[31] Sari, L., & Sari Siregar, G. Yanti Kemala. (2021). Perancangan aplikasi pendataan data kepegawaian negeri sipil pada Dinas Komunikasi dan Informatika Kota Metro. *Jurnal Mahasiswa Ilmu Komputer*, *1*(2), 115–135. https://doi.org/10.24127/.v2i1.1235

[32] Ramadhan, M., & Gustalika, A. (2024). Rancang bangun aplikasi pemesanan tiket tempat wisata berbasis Android menggunakan metode extreme programming. *Bulletin of Information Technology (BIT)*, *5*(2), 114–124. https://doi.org/10.47065/bit.v5i2.1341

[33] Priandika, A. T., & Riswanda, D. (2023). Perancangan sistem informasi manajemen pemesanan barang berbasis online menggunakan pendekatan extreme programming. *Jurnal Ilmiah Computer Science*, *1*(2), 69–76. https://doi.org/10.58602/jics.v1i2.8

[34] Nurdin, D. A., Firmansyah, D., Hafifi, A., & Darwis, M. (2024). Smart-working and hot desking application development using agile and extreme programming method at Xtra Cowork. *JISA (Jurnal Informatika dan Sains)*, *7*(2), 140–147. https://doi.org/10.31326/jisa.v7i2.2114

[35] Miracle, N., & Wibawa, R. P. (2024). Implementasi arsitektur microservice pada website business consolidation simulation menggunakan metode extreme programming. *Journal of Informatics and Computer Science (JINACS)*, *6*(02), 493–500. https://doi.org/10.26740/jinacs.v6n02.p493-500

[36] Pratama, S. D., Lasimin, L., & Dadaprawira, M. N. (2023). Pengujian black box testing pada aplikasi Edu Digital berbasis website menggunakan metode equivalence dan boundary value. *J-SISKO TECH (Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD)*, *6*(2), 560. https://doi.org/10.53513/jsk.v6i2.8166.