Development of Financial Planner Application Software Based on Waterfall Model

1st Jason Imanuel
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
jason.imanuel@binus.ac.id

2nd Lusia Kintanswari
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
lusia.kintanswari@binus.ac.id

3rd Vincent
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
vincent051@binus.ac.id

4th Maria Susan Anggreainy*
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
maria.susan001@binus.ac.id

5th Sablin Yusuf
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
sablin@binus.edu

6th Sula Yosege Sembiring Kembaren
Postgraduate Program
Bakrie University
Jakarta, Indonesia 12940
ORCID ID: 0000-0003-2738-7359

2022 International Conference on ICT for Smart Society (ICISS) | 978-1-6654-7135-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICISS55894.2022.9915039

***Abstract*— Financial planning is necessary, especially for those who started to have savings. It can be achieved through thorough planning, organizing, directing, and controlling financial activities. To overcome these problems, we developed and designed the financial planner app. It can help all our users to control their money flow according to their budgets every month. In developing this project, Business Model Canvas is used to assess the key points and Waterfall Model is used to define the process. All requirements of the financial planner app are modeled using Use Case Diagram. To cope with the reduplication of coding for the app development, we used the Object-Oriented approach and made a Class Diagram to portray the relation between objects involved. Besides that, we also use Sequence Diagram to develop the flow process of our system. We also developed a prototype of the User Interface to give a picture of how the app's visuals. Finally, we used the Black Box Method for testing the "Financially" app and the results are given in the form of tables representing every page. We hope that this "Financially" app will be helpful and the impacts on our users align with the purpose of the creation of the app.**

Keywords— financial, planner, application development, waterfall model, money.

I. INTRODUCTION

Financial planning is very important to be learned by everyone to be able to achieve their financial goals. [1] These financial goals could only be achieved by planning, organizing, directing, and controlling their personal financial activities. [2] One of the problems is that young people these days are struggling to achieve their financial goals because of the materialistic cultures around them. They tend to concentrate more on gaining assets rather than building financial plans. [3] The other problem that people are claiming that this financial planning is hard is that the financial services are because they have their own problems such as lack of trust, lack of investment funds, and the services have high fees or costs. [4] Not only the person that is affected by this financial planning problem but in the business, this financial planning is needed to fulfill the financial goals of the company. [5] The global financial crisis also led to the lack of trust of people in the financial planning industry. [6] There are some crucial reasons why financial planning is needed by everyone such as protecting our families from unpredictable risk, reducing debts, paying expenses to raise a family, having emergency funds, determining the type of life we would like to live after retirement, and many more. [7] To overcome these problems, we designed the financial planner application. It is an application that could help the user to control the money flow of their daily budget for a month. The financial planner has some important and useful features such as budgeting, making a summary of the user's money flow for a month, saving the information of the money spent by the user, and giving notifications regarding the budget that hasn't been used for the current time.

Our application aims to be helpful and simple to be used by users of all ages. Here are the few goals that become the focus of our application:

A. Helps in financial planning

We want to help those who can't control their finances. With our application, we hope that we can help people manage their finances better and save money for the uncertain future.

B. Ease in reviewing finances

People usually had a hard time remembering where all their money got spent on. With our application, we want to provide a detailed monthly financial report so that the users can keep a close look at their monthly spending.

C. Useful features to support user's financial plan

We enable the feature where users can segregate their funds into a bunch of budget plans. For example, a user can split his or her funds into fractions where each fraction has its own percentages from the total monthly budget. A user can set aside 50% of his total monthly budget for food and beverages, then 10% for donations, 20% for investing, and 20% for savings. This way, each user can become cautious whenever they overspend on one budget plan which shouldn't be done otherwise.

This application is developed by using the Waterfall model and Object-Oriented approach. Waterfall model is the breakdown of project activities into multiple phases which is linear and sequential. The most common characteristic that defines this model is that each phase is dependent and corresponds to the previous phases. This is widely used due to its model being simple and sequential, so it becomes step-by-step progress towards the goal of the project. However, a

978-1-6654-7135-0/22/$31.00 ©2022 IEEE

change made in one of the phases will alter the phases next to it, this becomes the downside of this model. This downside makes this model not the best option for projects that require a lot of revisions and changes since it will waste a lot of time. [8] Waterfall or structured analysis and design Model has sequential of steps of the development process and the data that is used in the system will be separated from the program that works on data. It also has rigid development steps from system and software requirements, analysis, program design, coding, testing, and operating. [9] [10]

II. METHODOLOGY

To develop this application, Business Model Canvas is needed to define process. Figure 1 shows the Business Model Canvas of this development. The methodology that is used to develop a Finance Application is represented in Figure 2. This methodology follows the waterfall model without deployment and maintenance for this research. Literature has also been done by reading to some journals and research about engineering and finance through web searches.

The first step to develop is analysis the problem and the objective of the application. In this case, the problem is that for some people financial planning or budgeting still hard to be managed. It also makes financial goals are troublesome to be achieved. And the objective of this research is to solve this problem, to make an application that can help people track their outcomes, and also budget their finance. Generally, this analysis step is done intensively to specify the needs of the systems so that they could be understood easily. [11]

KEY PARTNER
- partnering with the cloud services provider
- partnering with advertising companies
- partnering with backend developers
- partnering with frontend developers
- partnering with customer service provider company
- partnering with some influencers to share information of our application

KEY ACTIVITIES
- registration to make an account
- log in to access the account
- input budget
- input transaction
- make financial report

KEY RESOURCES
- 24/7 customer service facilities
- accessible anytime
- simple to be used for any generation
- free to download the application

VALUE PROPOSITION
- application for a personal financial planner by using the budgeting method
- the report can be seen and exported to an excel file so the user can track their money spending
- the user doesn't have to make a digital banking account but still can use the application

CUSTOMER RELATIONSHIPS
- Self-service
- Users can download, register, and create their own budget plans without needing permission.
- Customer Service
- Users can communicate with our staff in case a problem occurred to ensure customer satisfaction

CHANNELS
- As for the advertising steps, we use some social media platforms such as Instagram, TikTok, youtube, or others to promote our application.

CUSTOMER SEGMENTS
- People with difficulties in managing their finances
- Our app aims to help people manage their finances better for a better future
- All ages and genders
- Everybody has a limited amount of money. Because of that, we need to manage our money wisely. We can achieve that using this Financial Planner app

COST STRUCTURE
- the monthly payment for cloud services
- salary payment for developers (frontend/backend)
- salary payment for advertising uses
- salary payment for customer service company worker
- bonuses for staff during special occasions
- back up funds for unpredicted situation
- payment for influencer advertising cost

REVENUES STREAM
- advertisement in our application while generating a financial report
- in app purchases for no advertisement
- in app donation for developers (tip box)

Fig. 1. Business Model Canvas for Financial Planner App

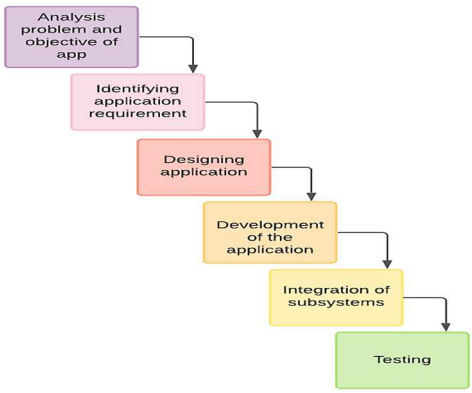

Fig .2. Flowchart for Finance App Development

Then we also need to identify the application requirement. All requirement will be used to build appropiate fitures, so we can satified what people need to do by using this application. For that reasons, by using object oriented approach, we define the use case, object, attribute, also the function to support the system. Then we will represent the requirement by designing the diagram in the next step. Generally, by finishing this step it would be resolved in a fully planned document or data that is needed by the application. [12]

To design an application, we need some diagrams to make the flow and connections of the application and the actors. The way the diagrams are implemented is by using UML (unified modeling language) to create a universal blueprint model. [13] The first type of diagram is the use case diagram that have the uses of showing the connection between the actor and the application, use case also features the feature of the application. The second type of diagram is the class diagram which is used to show the object, class, and variables

used in creating the application. The third diagram is the sequence diagram, this type of diagram is used to show the sequences of the application from the beginning the user uses the application until the user closes the application.

The following step is that we need to develop the application. In this step, we focus on planning the sequences of the software application. The type of application development that is being used in this software application is the waterfall method. Waterfall method is one of the methods that is used while planning the software application. By using this waterfall method, it is presumed that the software engineers are fond and clear of the application that they are meant to create. Because by using this method, all steps are being finished step-by-step so that there are no going back steps. Everything is done step-by-step so that the only thing that is presented to the user is that the final project which is the software application. This waterfall method is better to be used by small project applications with clear meaning and functions, rather than a big project which requires many revisions. Development step is the step where the codes are written to fulfil the functionality of the application, sometimes this step is being broken down into some divisions to ensure that all code problems are being handled more easily. [14] The process of coding in this application is where the conversion from the processing phase into the production phase happens. [15]

To piece together an application, there will be a lot of components involved that need to be integrated. This is necessary in order for the application to be able to have better performance and perform a wider scope of tasks. This process involved linking together all of the subsystems so that they can work functionally as one complete system. [16] For the method of integration, point-to-point integration is used if the application involves only two subsystems, the front end, also the back end. The front end is the User Interface and the back end would be the code done in Object-Oriented Programming language. In general, this process is needed for the implementation of a good planning process for the application or system. [17]

Once all of the steps above are done, it is crucial to test the application. This is done for three main purposes: identifying defects, reducing flaws, and increasing the overall quality of the application. [18] Defects include those bugs and errors caused by human errors. [19] Reducing flaws means removing unnecessary things that might lower the quality of the application. Increasing the overall quality can be done by identifying which parts can be improved and adding them for perfecting the application. Generally, unit testing, integration testing, validation testing, and system testing are the most common type of testing. [20]

III. III. DESIGN OF DEVELOPMENT

A. Modelling of application requirement

All requirements of the Financial Planner Application are modeled by using a *use case diagram* that can be seen in Fig 3.

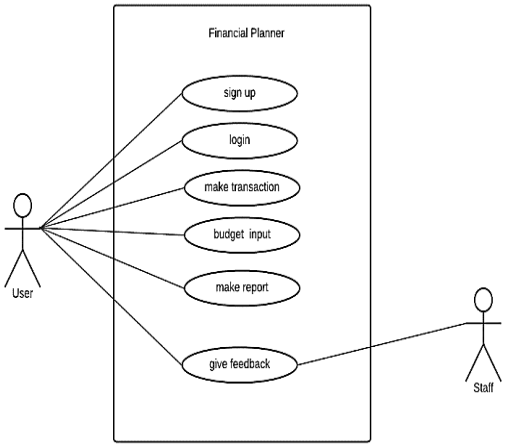

Fig 3. Use Case Diagram of the Financial Application

Our use case diagram has 2 actors which are the user and the staff. The user is the one who uses the application, and the staff is the ones who could help the user when a problem occurs. Our use case diagram has many use cases:

- Sign up feature, which allows the user to register their own account.
- Login feature, which allows the user to log in to their account so that their money flow becomes confidential information and can only be managed and seen by the user itself.
- Setting display, which allows the user to design the display according to their desired style
- Budget input feature, which allows the user to input their monthly budget so that the system could do the budgeting steps from the user's monthly budget.
- Make a report feature, which allows the user to get their monthly report of their money flow.
- Give feedback, the user will use this feature whenever a problem occurs, and this feedback will be given to the staff to be solved.

B. Class Design

To cope with the reduplication of coding for application development we must design the relation of the object and use an Object-Oriented approach. [11] All identification of all classes is presented by using the *Class Diagram* in Fig 4.

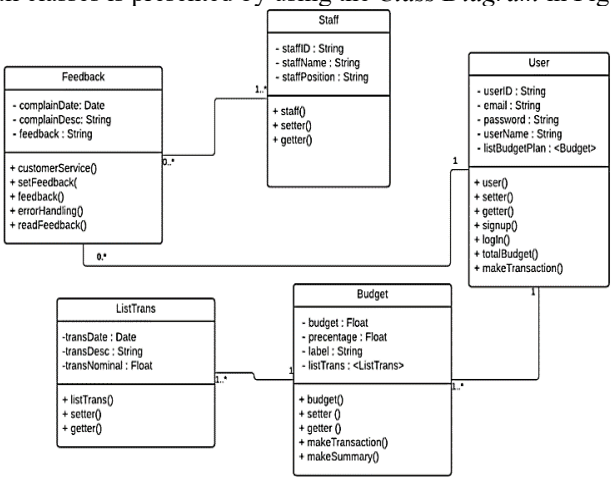

Fig 4. Class Diagram of the Financial Application

Our class diagram has multiple classes to support our application. We will briefly explain what each class is for:

- Feedback: contain the date, description, and resolution of the complaint/problem.
- Staff: contains the information regarding the staff managing our application.
- User: contains the information about the users that use our application.
- ListTrans: contains the transaction date, description, and amount.
- Budget: contains the total monthly budget, the percentage of each budget plan, the label of each budget plan, and the list of budget plans that the user has created.

C. *Scenario Design*

Our application starts at user login, in the condition of no account found, the user will be redirected to register an account and there exists a loop that repeats the whole registration process when there are errors in the information given by the user. Once an account is made, the user will be given the opportunity to set their own budget plans which can be more than one. Next, the user can add transactions with labels and the amount of money spent on each transaction. In order to generate a monthly financial report, objects from the class ListTrans will be retrieved by using the getter function and it will continue to loop until all of the data has been received. The monthly transaction summary will be generated for the user to review. Lastly, in the case where the user has a problem while using the application, the user can contact our customer service. Our staff is going to handle the problem submission and give a reliable solution to the problem.

D. *User Interface Design*

A. Loading page is a page the user sees when the application is on load. This page is represented in Fig 5. The login page is where the user fills in his/her email and password in order to log into his/her Financially account. This page is represented in Fig 6.

Fig 5. Loading page

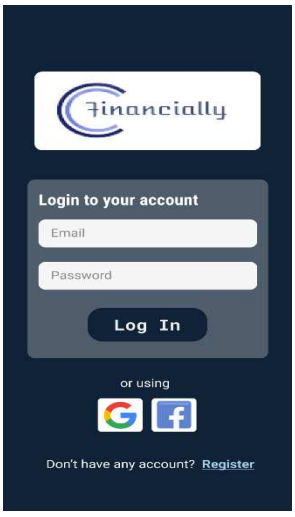

Fig 6. Login page

Fig 7. Registration page

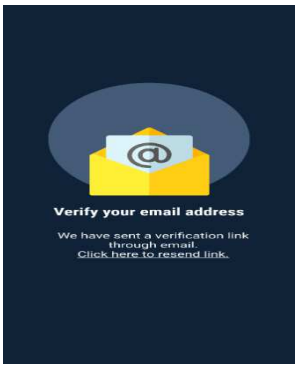

Fig 8. Verification page

The registration page is where the user can fill in his / her information necessary for the creation of the account. This page is represented in Fig 7. This is the verification page where User is required to verify their account by clicking the link sent to their email address. This page is represented in Fig 8.

Fig 9. Home Page

Fig 10. Customer service page

This is the dashboard or the home page of the Financially App. It displays the User's name, the preview of the ongoing budget plan, and the other four features available. This page is represented in Fig 9. Customer Service page where the User can contact our Admin in case of a problem with the app or an uncertainty about the app's features. This page is represented in Fig 10.

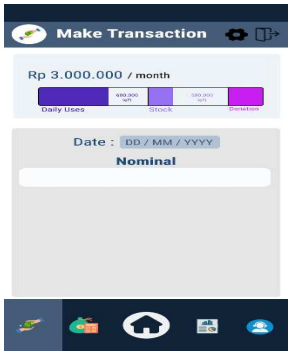

Fig 11. Make transaction page

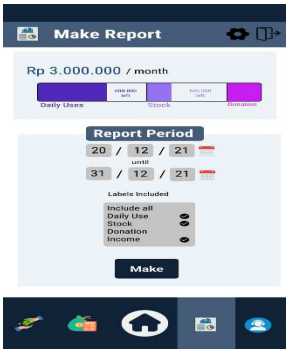

Fig 12. Make report page

Make transaction page is where to user can instantly record a transaction and assign it directly to the designated budget plan. This page is represented in Fig 11. Make a report

page where users can generate a report based on their transactions, which the user can select to include all the labels or only the desired few. This page is represented in Fig 12.

Fig 13. Transaction history page

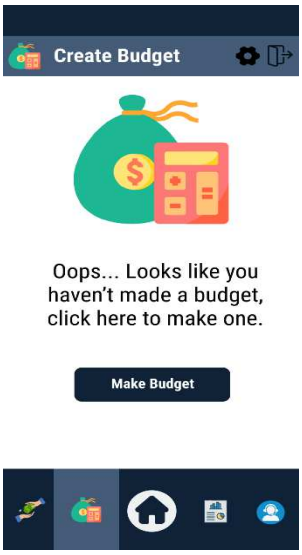

Fig 14. Create budget page

This is the second page of the make report which shows a preview of the report and gives the user the file format of the report. This page is represented in Fig 13. Create Budget page is available if the User wanted to create a new budget plan or simply edit and modify the current budget plan. This page is represented in Fig 14.

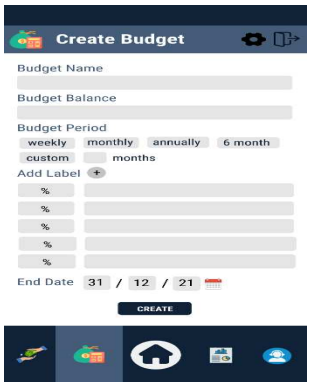

Fig 15. Create budget page

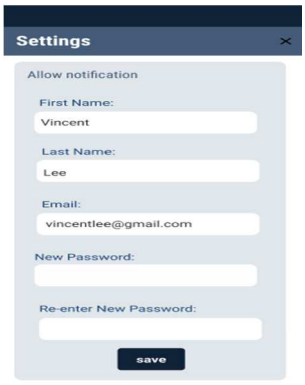

Fig 16. Setting page

The second step of creating budget page asks the user to fill in the identifiers of the budget plan such as name, balance, period, labels included, and the duration of the budget plan.

IV. TESTING

The testing of the financially application is using black box method. The results are represented by the following tables Table I, Table II, Table III, Table IV, Table V, Table VI, Table VII, and Table VIII. The test results show that all functions have been running successfully

TABLE I. TABLE TESTING OF LOGIN PAGE

No	Process	Response	Status
1	User clicks login button	System moves to the homepage	Success
2	User clicks the register button	System will move to the registration page where User can input his/her personal data to get registered	Success

TABLE II. TESTING OF THE REGISTRATION PAGE

No	Process	Response	Status
1	User inputs his/her personal registration information	System displays the inputted data from the respective boxes	Success
2	User ticks the agreement to the terms and policy box	System displays that the User has agreed on the terms and conditions	Success
3	User clicks the register button	System moves to login page	Success

TABLE III. TESTING OF HOME PAGE

No	Process	Response	Status
1	Users click the *homepage logo* from the navigation bar	System displays the User's budgeting plan and the amounts of the expenses	Success
2	User click *edit logo* from the "home" page	System shows the page to edit the budget	Success
3	Users click the *setting logo* from the "home" page	System displays the setting menu	Success
4	User click the *back logo* from the "home" page	System displays the "log in" page as the User is logged out	Success

TABLE IV. TESTING OF CUSTOMER SERVICE PAGE

No	Process	Response	Status
1	User click the *customer service* logo from the navigation bar	System displays Customer Service's opening bubble chat to User	Success
2	User click the *customer service* logo from the navigation bar	System displays Customer Service's opening bubble chat to User	Success
3	User slides the slider of the message.	System displays the message between the User and the Customer Service as the User slides.	Success

TABLE V. TESTING OF TRANSACTION PAGE

No	Process	Response	Status
1	User click the *transaction* logo from the navigation bar	System displays Make Transaction page to User	Success
2	User click and choose the current date of the transaction	System displays the date, month, and year for the User to choose	Success
3	User click and insert the nominal of the transaction	System displays the nominal of the User's transaction	Success
4	User click and type the description of the transaction	System displays the description of the User's transaction	Success
5	User choose and click the categorization of the transaction	System categorized the User's transaction detail	Success
6	User click the *add transaction* button	System saves the User's transaction detail	Success

TABLE VI. TESTING OF REPORT PAGE

No	Process	Response	Status
1	User click the *report logo* from the navigation bar	System displays Make Report page to User	Success
2	User click and choose the starting date for the report	System displays the starting date for the report	Success

3	User click and choose the end date for the report	System displays the end date for the report	Success
4	User click the label of the report	System displays the checklist for the report	Success
5	User click the make button	System displays the User's transaction history report	Success

TABLE VII. TESTING OF SET BUDGET PAGE

No	Process	Response	Status
1	Users choose to *Create Budget* Page	System shows the budget list if there exists, or shows only a message if User hasn't made any budget plan	Success
2	Users choose the *Make Budget* button	System goes to another slide, to ask the User the budget plan	Success
3	User input the *Budget Name*	System gets the *Budget name*	Success
4	User input the *Budget Balance*	System gets the *Budget Balance*	Success
7	User set the *End Date*	System set the end date of the budget	Success

TABLE VIII. TESTING OF SETTING PAGE

No	Process	Response	Status
1	User chooses *Setting* button	System shows *Setting* page	Success
2	Users change the *First Name* and *Last Name*	*First Name* and *Last Name* are changed	Success
3	Users change the *Email*	*Email* is changed	Success
4	User input the *New Password*	*New Password* is inputted	Success

V. CONCLUSION

Seeing the current pace of growth in technology results to faster transactions between two entities. This ease in the transaction also affected people psychologically, making them easier in spending their savings without thinking ahead. Financially helps to assist people in their financials, helping them create the boundaries in limiting their outflows. Financially allows its users to create budget plans in order to control and monitor their spending and incomes every month. The savings gained are meant to provide emergency funds in case something bad happens in the future. Financially aims to help the users develop a good habit to be wiser and avoid being consumptive on luxury needs to gain status. We conducted testing on the app and all of the features, the result of it was a success and the system responded correctly to every action the user made. We believe that the development of this app will be helpful to all its users, as mentioned before, users from junior high school and older who started having savings. We hope that the impacts to our users align with the purpose of the creation of the Financially app.

ACKNOWLEDGEMENT

This work is supported by Research and Technology Transfer Office, Bina Nusantara University as a part of Bina Nusantara University's International Research Grant (PIB 2021) with contract number: 017/VR.RTT/III/2021.

IV. REFERENCES

[1] Dwi Ningrum, Hafida Alviolita (2021). Faktor yang Memengaruhi Financial Planning Pada Masyarakat Kota Kediri. *Jurnal Ilmu Manajemen, Volume 9 No 2, 2021.*

[2] Grozdanovska Vesna, Bojkovska Katerina, Jankulovski Nikolche (2017). Financial Management and Financial Planning in the Organization. *European Journal of Business and Management, Volume 9 No 2, 2017.* https://core.ac.uk/download/pdf/234627714.pdf

[3] Pareek Nishtha, Dixit Neerja (2016). Financial Literacy and Attitudinal Study of the Newly Employed Youth towards Financial Planning. *Social Sciences ISSN, Volume 4 Issue 2, 2016, 358-370.* http://dx.doi.org/10.21013/jmss.v4.n2.p5

[4] Chira Inga (2016). Perception about the Financial Planning Profession. *Journal of Wealth Management 18((4):22-28.* 10.3905/jwm.2016.18.4.022

[5] Snider, J.H., Danies, K.J. (2018). Success Strategies for Small Finance Planning Firm. *International Journal of Applied Management and Technology, Volume 17 Issue 1, 22-34.* 10.5590/IJAMT.2018.17.1.02

[6] Gunardi Ardi, Ridwan Mochammad, Sudarjah, G. M. (2017). The Use of FInancial Literacy for Growing Personal Finance. *Jurnal Keuangan dan Perbankan, 21(3): 446-458, 2017.* http://jurnal.unmer.ac.id/index.php/jkdp

[7] Alisyahbana, S. Rozy, N.F. (2017). Aplikasi Perencana Keuangan Pada Platform Berbasis Smartphone Multiplatform Dengan Framework Phonegap. Jurnal Teknik Informatika, 10(1), 67-72. doi:10.15408/jti.v10i1.6818.http://journal.uinjkt.ac.id/index.php/ti/article/view/6818

[8] Tutorials Point organization, India, accessed 23 November 2021, https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm >

[9] Adetokunbo A.A. Adenowo, Basirat A. Adenowo (2013). Software Engineering Methodologies: A Review of the Waterfall Model and ObjectOriented Approach. *International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July 2013, 427-434.*

[10] Wilfred van Casteren. (2017). The Waterfall Model and the Agile Methodologies: A comparison by project characteristics.

[11] Andriansyah, Doni. (2018). Penerapan Model Waterfall Pada Sistem Informasi Jasa Laundry Berbasis Web. *Indonesian Journal on Software Engineering, Volume 4 No 1,2018.*

https://ejournal.bsi.ac.id/ejurnal/index.php/ijse/article/view/6291

[12] Tujni B., Hutrianto, H. (2020). PENGEMBANGAN PERANGKAT LUNAK MONITORING WELLIES DENGAN METODE WATERFALL MODEL. *Indonesian Journal Ilmiah Matrik, 22(1), 122-130.* https://doi.org/10.33557/jurnalmatrik.v22i1.862

[13] *Prasetyo, E., & Putra, A. (2021). Implementasi Waterfall Model Dalam Pengembangan Sistem Informasi Eksekutif Penduduk. Journal of Information Systems and Informatics, 3(1), 213-224.* https://doi.org/10.33557/journalisi.v3i1.121

[14] *Eason, Oriana Karina, "Information Systems Development Methodologies Transitions: An Analysis of Waterfall to Agile Methodology" (2016). Honors Theses and Capstones. 286.* https://scholars.unh.edu/honors/286

[15] *Aroral, Harkirat Kaur. (2021). Waterfall Process Operations in the Fast-paced World: Project Management Exploratory Analysis (2021). International Journal of Applied Business and Management Studies. Vol. 6 No. 1, 2021.*

[16] *Lehtonen, K. (2018). What is system integration?* https://www.youredi.com/blog/what-is-system-integration

[17] *Nugraha, Yopi. (2020). JURNAL RISTEC: Research in Information Systems and Technology Vol. 1 No. 2. 2020.*

[18] *Sharma, L. (2021). Why is Testing Necessary and Important?* www.toolsqa.com/software-testing/istqb/why-is-testing necessary

[19] Thomas Hamilton (2021). What is Software Testing? Definition, Basics, & Types in Software Engineering.

[20] *Sara Alshathry, Betool Alnamlah, Norah Alkassim, Nor Shahida Mohd Jamail. (2020). Risk Management in Agile and Waterfall Models: A Review. International Journal of Advanced Science and Technology, 29(9s), 1149-1157.*