SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 2, June 2025, pp.365 - 377
ISSN 2407-0939 print/ISSN 2721-2041 online

Manufurrt: A Feasibility Investment and Operational Desktop Calculator

Kurniawan Hamidi[1*], Handi Wilujeng Nugroho[2]
[1,2] Department of Industrial Engineering, Faculty of Technology, Universitas Universal
Jl. Pasir Putih, Kelurahan Sadai, Kecamatan Bengkong, Kota Batam, Kepulauan Riau 29432
Email: khmd93@gmail.com, handynugroho41@gmail.com

ABSTRACT

The prospect of success of a manufacturing project is determined by the results of feasibility tests such as investment and operations. In its implementation, the feasibility test is often constrained by several things such as data entry errors, calculation errors, and a long calculation time. The ideal solution that can be used to solve these three obstacles is to design an offline and stand-alone desktop-based feasibility calculation application. This research uses the SDLC Waterfall method and User Acceptance Test (UAT) based on questionnaires and scenario tests. In the design and coding completion phase, the SDLC waterfall approach is combined with generative AI to speed up the application design time. As a result, a calculation application named Manufurrt has been created with an interactive executable file format and is able to provide the same calculation results between the expected spreadsheet and the real calculation results from the application.

Keywords: *Custom tkinter; Desktop App; Feasibility Analysis Calculator; Python*

Introduction

Project feasibility analysis is a crucial step in determining the prospects of a project's success[1]. In the context of manufacturing projects, the process of making optimal investment decisions is often faced with obstacles due to limited analytical skills and understanding of financial and operational feasibility indicators, such as Net Present Value (NPV), Break-Even Point (BEP), Cost of Goods Manufactured (HPP), Payback Period (PBP), and Operating Costs (BOP). This phenomenon is common in project management practices in various sectors, especially when calculations are done manually or through spreadsheets that are prone to input errors, interpretation errors and time efficiency issues[2][3][4][5][6]. As the complexity of investment projects increases, the urgency related to the availability of analytical tools that are able to facilitate the evaluation of financial feasibility in a systematic, accurate, and efficient manner for cross-disciplinary stakeholders has become increasingly urgent [7][8][9]. This need is explicitly in line with the making Indonesia 4.0 programmed based on PERPRES No. 74 of 2022 concerning national industrial policy with one of the main focuses being digitalization in various industrial sectors. Implicitly, the need is also in line with the policy of PERPRES No. 38 of 2015 regarding the necessity of thorough feasibility studies in investment projects, especially public infrastructure projects.

The ideal solution to support such digitized investment decision-making is the development of a stand-alone application system capable of integrating various investment project calculations on a single platform. The system is equipped with a simple Graphic User Interface (GUI) but is able to display comprehensive and accountable results[10][11], allowing users to adapt to the system quickly, navigate the system efficiently, and minimize potential calculation errors[12][13][14].

Various approaches to digitalization have been adopted. This research[15] focuses on developing a stand-alone FiNA-based mobile application with the Dart language and Flutter GUI framework to simplify and accelerate the process of analyzing the feasibility of financial investments using API data integration from Bloomberg. In a similar research on *Mobile Feasibility Applications* [16], FiNA is used to make informed decisions for consultants and developers, so the application allows users to test various proposals and observe the results in real-time and improve users' understanding of feasibility analysis. Based on research [17], utilizing automated workflow development with command-line Python scripts to integrate economic models with spreadsheets, dynamic simulators, and uncertainty analysis tools in an iterative manner for financial simulation and evaluation processes to shorten decision-making time. Other research [18], implements financial feasibility analysis on prefabricated production networks and injection molding equipment to test the feasibility of investing in modular production equipment and systems in large-scale manufacturing projects using spreadsheets integrated with Python. Most research relies on platforms such as conventional spreadsheets, highly-licensed commercial software to perform simulations and mobile-based applications [15][16][17][19][20][21][22]. In addition, most of the approaches developed have not provided a stand-alone, offline desktop application system to comprehensively analyze investment and operational feasibility.

Based on this, this research proposes the creation and development of a prototype Python-based desktop application to facilitate users when calculating and interpreting data offline. The application is specifically used to calculate investment and operational feasibility metrics, including NPV, PBP, BEP, HPP, and BOP and offers scenario and sensitivity analyses to support the decision-making process and risk evaluation process.

The development team for this research consists of academics and practitioners in industrial and manufacturing engineering with experience in developing Python-based applications and a deep understanding of engineering economics methodologies. These competencies underpin the development of applications that are not only technically valid, but also adaptive to the needs of users in the field. The developed application is expected to be utilized as a medium to support the learning, planning, justification, and decision-making processes related to investment feasibility at the operational and strategic levels, especially in specific contexts related to manufacturing projects.

Research Methods

This research uses a comprehensive software engineering methodology, namely the waterfall Software Development Life Cycle (SDLC) for building desktop applications with the testing stage using Black box testing to test the functionality of the code based on the suitability of input and output without looking at the structure of the program code[23] and User Acceptance Test to test the feasibility of the system from the end-user's point of view[24]. The Waterfall SDLC method is a traditional software development methodology characterized by a linear and sequential approach. This methodology involves different phases such as requirement analysis, design, coding, testing, installation, and maintenance. Each phase must be completed before moving on to the next phase[25][26][27][28].

Requirement Analysis

This stage is the most important stage because this stage is used to clarify and define the problem[29]. The aim is to ensure that the design phase runs smoothly[30]. At this stage, all needs or system requirements from users and or stakeholders are collected and documented in the software requirement specification (SRS)[31] and/or written documentation in the form of a requirements specification metric[32]. Challenges that often occur in this phase are differences in goals between stakeholders, misunderstandings between stakeholders and developers, ambiguity, dynamics of user desires and psychological differences between professionals that can lead to a lack of accuracy and completeness of data[33]. Therefore, Ontology-based mapping of different requirements and the application of Unified Modelling Language (UML) patterns are used to improve the clarity, accuracy and completeness of requirements documentation[33][34].

System Design

The design process begins with designing a comprehensive system architecture that includes determining the overall shape of the system, identifying the components used, understanding the interactions between components, and determining the needs of the tools needed to support the implementation of the application[35]. In the process, this phase can be integrated with generative AI technology to make the design process easier, faster and more productive[36].

Implementation/Coding

The coding phase is the core phase in the creation and development of desktop applications. In this phase, users often face repetitive tasks such as method definition tasks, GUI component creation tasks such as buttons, labels, entry fields, widget component dimensioning tasks and looping tasks for error handling and debugging[37].

Black Box Testing

This phase is a phase of testing the functionality and testing the output of the application without knowing the code structure. This phase uses equivalence partitioning, boundary value analysis, decision table testing in the case design test. The next test is test execution, which is the process of comparing the actual output with the expected output, then ending with analyzing the output error using generative AI to clarify the error condition even without seeing the code.

User Acceptance Testing (UAT)

UAT is a stage of testing the feasibility of applications by end users which is carried out in 3 ways, namely: assessing the suitability of the application to the initial requirements, assessing the functionality of all features and ensuring there are no errors when the application is used. This study used 3 end users with a total of 25 test scenarios for all metrics.

Table 1. UAT testing scenario

Methods	Amount of data
NPV	7 data (normal, edge, error)
PBP	6 data
BEP	5 data
HPP	5 data
BOP	2 data
Total	25 data

Table 2. Questionnaire for UAT assessment

Statements	Scale	Description
The app is easy to use (P1)	1-4	1: strongly disagree, 2: disagree, 3: agree, 4: strongly agree
The calculation result is correct (P2)	1-4	
The data input process is clear (P3)	1-4	
Analysis scenario feature is useful (P4)	1-4	
I will use it in the real case feasibility calculation process (P5)	1-4	

To strengthen the empirical foundation of this study, several case examples, and findings from recent empirical studies in [38], [39], [40] are incorporated. These data provide a contextual validation of the proposed Manufurrt application in real-world settings. The UAT results serve as primary empirical evidence, while additional references offer supporting secondary data to reinforce the practical utility of the application.

Deployment/Installation

This phase is the process of distributing and installing a Windows-based desktop application in the form of an executable (.exe) file format to the end-user's computer memory[41]. The executable file is the final result of the coding and compile process on the desktop application so that the end-user can run it independently on a computer without the internet[42].

Results and Discussion

Based on the sequence in the previously designed research method, the research results are analyzed and discussed in this section.

Requirement Analysis Results

The needs identification process is mapped based on Ontology. Ontology is a formal structure that describes the relationship of concepts in a particular domain in a system. In this case, the ontology consists of investor, investment, calculation standard, manufacturing project, investment feasibility and operational feasibility. Investors act to make investments with a certain investment value, and these investors make several calculation standards as a basic consideration for the feasibility of investing and operating a manufacturing project. On the other hand, these investors experience constraints such as time to decide and are faced with a lot of data input to be calculated, therefore, the data needs are documented in Table 3. based on interviews with academics and practitioners in related fields.

Table 3. User needs & spesification metrics

No	Needs	Metrics
1.	need a simple investment feasibility calculation application (NPV, PBP, BEP) and compatible without having to use the internet	Executable file (.exe) and offline with execution time < 2 seconds
2.	need a simple and compatible application to calculate operational feasibility (COGS and BOP) without having to use the internet	Executable file (.exe) and offline with execution time < 2 seconds
3.	Features a customized menu for each calculation indicator	There is > 1 main menu for each investment feasibility and operational feasibility
4.	The application can display an error notification when the input entered in the field is incorrect.	There is an error message notification and a notification to fix the input again
5.	Each menu has inputs that can be customized with user input and display the right outputs.	All input fields can accept a range of values according to the provisions of the indicator and all outputs produce valid calculation output
6.	has a reset feature on each indicator and a back to menu feature	Has reset and back to menu buttons on every page with each process time < 1 second

SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 2, June 2025, pp.365 - 377
ISSN 2407-0939 print/ISSN 2721-2041 online

UML: Activity diagram

Unified Modeling Language (UML) is a standard diagram to visualize the structure and process of how a system works. One of these UMLs is the activity diagram. The activity diagram in this case is described in the following description and shown in Figure 1. The user runs manufurrt.exe, then the system of the application will display the main page which has a logo and ribbon bar. On the ribbon bar there are menu, help and exit options. The menu bar serves to direct the user to the menu page which consists of all the eligibility indicators discussed in the research. In addition, on the menu page ribbon bar, users can choose to return to the main page or exit the application by pressing the home button and exit button on the ribbon bar. If the user selects one of the calculation indicators, the user is directed to the calculation page. Each calculation indicator page has a calculate button, reset button, input and output fields, menu bar and exit bar. If the user has finished calculating, the application can be closed by pressing the exit button on the exit bar.

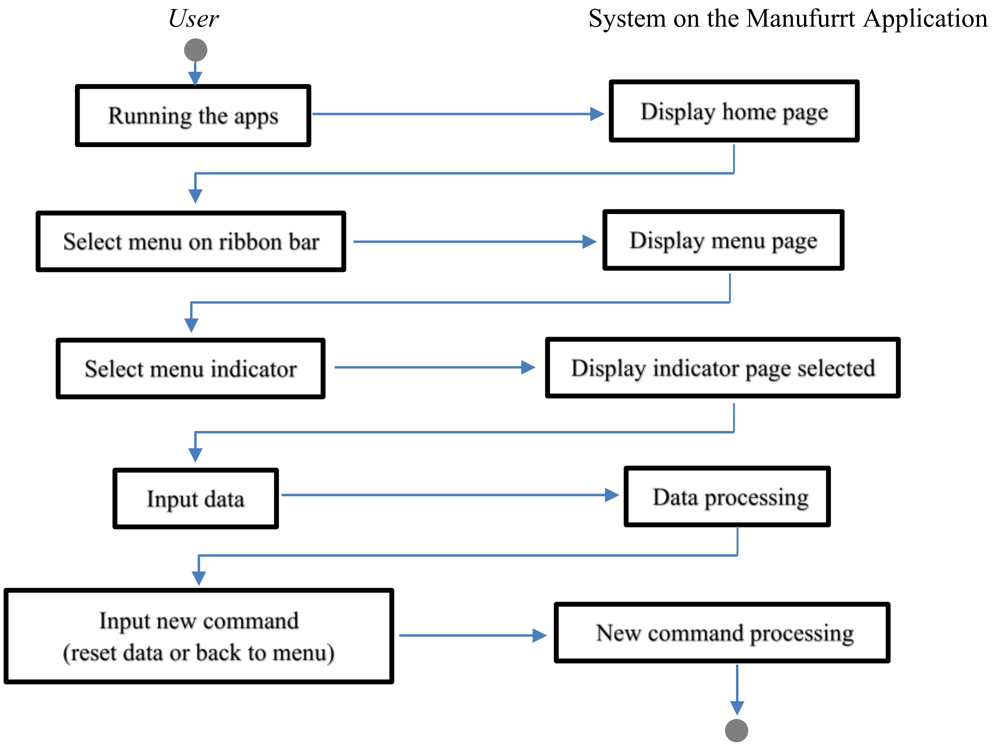

Figure 1. *Activity diagram*

System Design: *High Fidelity*

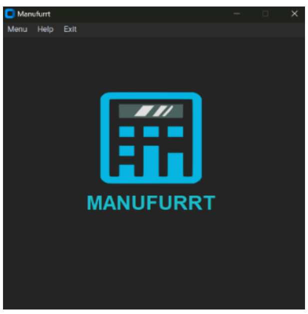

Figure 2. Home page

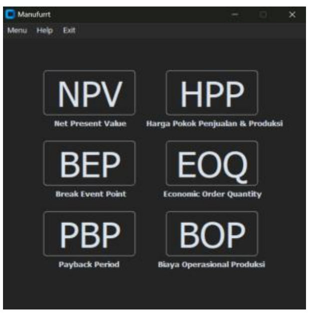

Figure 3. Menu page

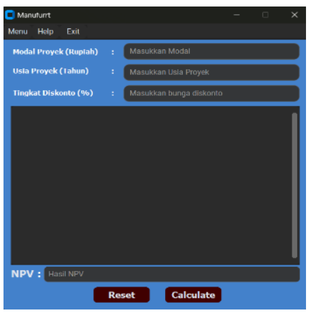

Figure 4. NPV page

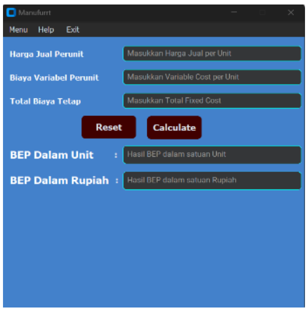

Figure 5. BEP page

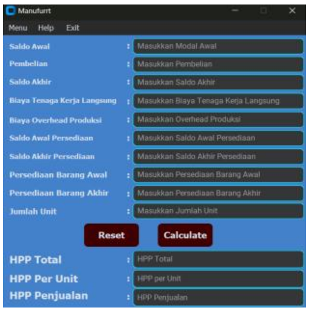

Figure 6. HPP page

Figure 7. PBP page

The GUI mockup display in Figure 2 to Figure 7 is a visualization that is close to the final appearance and was designed using Canva.

Implementation of Coding

During the system design and coding phase, generative AI tools particularly ChatGPT 4.0 were utilized to assist in automating code generation, GUI layout suggestions, and user input validation logic. While these tools significantly accelerated the prototyping process, several technical challenges were encountered.

First, prompt sensitivity and output inconsistency often resulted from minor variations in query phrasing, leading to non-uniform code structures. This required manual curation and standardization to maintain coding consistency. Second, the generated code occasionally contained syntax and logical errors, including outdated method calls, missing imports, and improper data binding mechanisms, which necessitated extensive debugging and refactoring. Third, there were integration difficulties, particularly in aligning AI-generated components with the modular architecture of the Python–Tkinter framework, especially in managing application states, callback functions, and event handlers.

To address these issues, the development team adopted a hybrid approach, wherein generative AI was primarily used for ideation, prototyping, and component-level suggestions, while human developers were responsible for validation, structural harmonization, and stress testing. Additionally, code modularization strategies were implemented to enhance the adaptability and maintainability of the application. Domain-specific GUI templates were also developed based on engineering design principles to ensure usability compliance.

These mitigation strategies not only enhanced the robustness and reliability of the software but also underscored the importance of human oversight in AI-assisted development, particularly in academic tools that require high levels of precision, traceability, and functional validity. This principle is further demonstrated in the implementation phase, where Figure 8 and Figure 9 present interface snapshots and corresponding code fragments used to calculate one of the key feasibility indicators, namely Net Present Value (NPV).

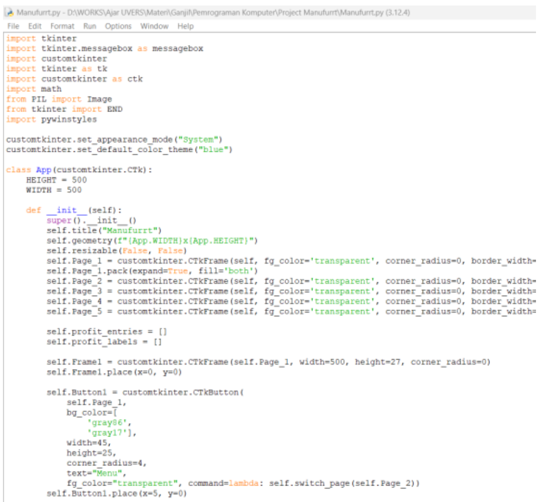

Figure 8. Part of Manufurrt codes

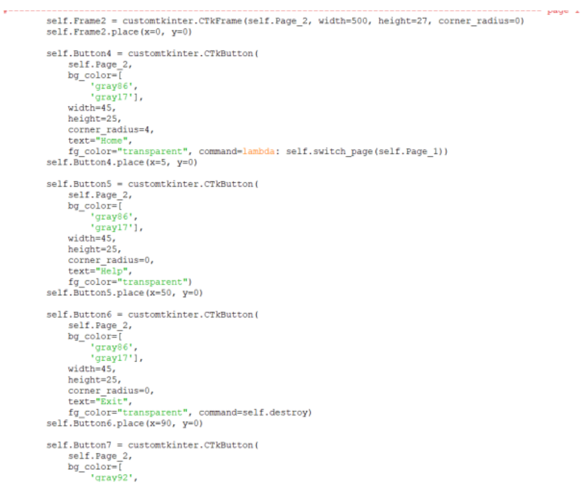

Figure 9. Part of Manufurrt GUI codes

Black Box Testing

Table 4. Recap of Black box test results

Functionalities	Description of Test Results
Home page	It works
Ribbon bar menu	It works
Ribbon bar exit	It works
Ribbon bar help	It works
Menu page	It works
Ribbon bar home	It works
NPV calculation page	It works
NPV calculation button	It works
Button for reset all NPV data	It works
Input error notifications for all calculation indicators	Appears/works
All input and output fields for each calculation indicator	Works, and displays input and output values
PBP calculation page	It works
PBP calculation button	It works
PBP reset data button	It works
BEP calculation page	It works
Buttons calculate BEP	It works
Button reset data BEP	It works
OGS calculation page	It works
HPP calculation button	It works
Button reset data HPP	It works
Page switch button	It works
Main BOP calculation page	It works
BOP rate calculation page	It works
Buttons calculate BOP rate	It works
Button to reset data rate BOP	It works
Button returns to the main BOP menu	It works
Direct labor BOP calculation page	It works
Buttons to calculate TKL BOP	It works
Button to reset TKL BOP rate data	It works
Raw material BOP calculation page	It works
Labor hour BOP calculation page	It works
Machine hour BOP calculation page	It works
Product unit BOP calculation page	It works
All calculate button, reset button, return button on BOP	It works

One example of proof of functionality test is the appearance of error notifications in the application for NPV and BEP calculations. In Figure 10 and Figure 11, the application shows an error notification in the output field when the calculate button is pressed. In Figure 10, the user tries to input a value of 0 in the discount, then the user presses the calculate button, so an error message appears in the output field. Whereas in Figure 11, the user tries to input something other than the number in the BEP calculation input field, so the output shows an error notification in the form of a message box when the calculate button is pressed.

User Acceptance Test

UAT testing is done using a questionnaire. End users were asked to provide a calculated test value on the 2nd statement in the questionnaire. To make the assessment unbiased, end-users were given 25 scenarios to be tested using the application and compared with the expected calculation using a spreadsheet. The detailed calculation scenarios are organized as follows:

1. Normal case NPV with initial investment input = 100,000; Discount rate = 10% and cashflow (30,000; 40,000; 50,000).
2. NPV Sensitivity Analysis with initial investment input = 100,000; Discount rate = 9%, 10%, 11%, and cashflow (30,000; 40,000; 50,000).
3. Edge case NPV with initial investment input = 100,000; Discount rate = 10%, and cashflow [] (empty cashflow)
4. Edge case NPV with initial investment input = 100,000; Discount rate = 10%, and negative cashflow (-10,000, -20,000, -30,000)
5. Error case handling for NPV with initial investment input = 100,000; Discount rate is empty, and negative cashflow (-10,000, -20,000, -30,000)

6. Error handling case for NPV with initial investment input= 100,000; blank discount rate, and sequence cash flows (30,000, a, 50,000)
7. Scenario Analysis case for NPV with initial investment input= 100,000; blank discount rate, and negative cash flows (-10,000, -20,000, -30,000)
8. Normal PBP case with initial investment input = 100,000 and annual average cashflow (40,000)
9. Normal case of PBP with initial investment input = 100,000 and annual cashflow (20,000, 30,000, 40,000)
10. PBP edge case with initial investment input = 100,000 and annual cashflow (110,000, 0, 0)
11. Error handling PBP case with input initial investment = blank and annual cashflow (30,000, 30,000, 30,000)
12. PBP normal case with initial investment input= 100,000 and average annual cashflow (40,000)
13. Sensitivity analysis case with initial investment input = 100,000 and annual cashflow variation of 20,000 to 60,000
14. Normal BEP case with input fixed cost = 50,000, selling price 500, and variable cost per unit 300
15. Edge case BEP with fixed cost input = 50,000, selling price 500, and variable cost per unit 0
16. Error handling case for BEP with fixed cost input = 50,000, selling price 300, and variable cost per unit 350
17. Sensitivity analysis case for BEP with fixed cost input = 50,000, selling price 400, 450, 500, and variable cost per unit 300
18. Normal case of BEP with fixed cost input = 50,000, and margin per unit 200
19. Normal case COGS with main input of material cost= 30,000, labor cost= 20,000 and overhead cost= 10,000
20. Edge case COGS with main inputs of material cost= 30,000, labor cost= 20,000 and overhead cost=0
21. Error handling case for HPP with the main input of material cost= 30,000, labor cost= blank and overhead cost= 10,000
22. Sensitivity analysis case for COGS with the main input of material cost variation (25,000, 30,000, 35,000) labor cost= 20,000 and overhead cost= 10,000
23. Normal case of HPP per unit with total input of HPP = 100,000 and production = 200 units
24. Normal BOP case with budgeted BOP input=50,000 and estimated expense basis=250
25. Error handling case for BOP with budgeted BOP input= 50,000 and estimated expense basis= empty

Based on the testing of 25 scenarios by each end user, it is found that all calculation test results in each scenario with the Manufurrt application are the same as the expected calculation test results based on spreadsheets. The proof of the test fragment can be seen in Figure 12.

Table 5. Questionnaire test results

End-user	P1	P2	P3	P4	P5	score	max score
1	3	4	3	4	3	17,00	20
2	4	4	3	4	4	19,00	20
3	4	4	3	4	3	18,00	20
			Total			**54,00**	**60**

Based on the results of questionnaire tests obtained from 3 end users, the average acceptance test score obtained is 3.60 and the percentage of feasibility of using the application is 90%. This shows that the application is considered feasible and acceptable by end-users to be used in the calculation of investment feasibility analysis and operational feasibility of manufacturing projects.

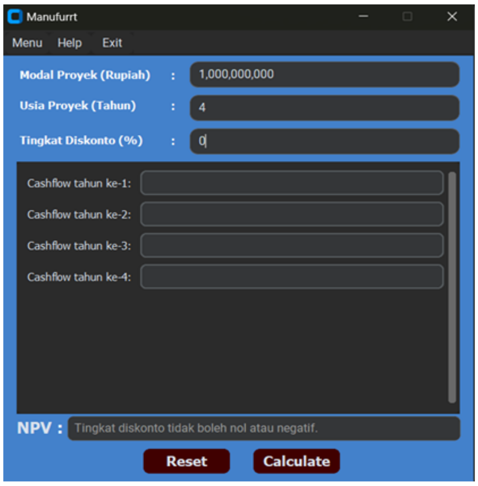

Figure 10. Error notification on NPV output field

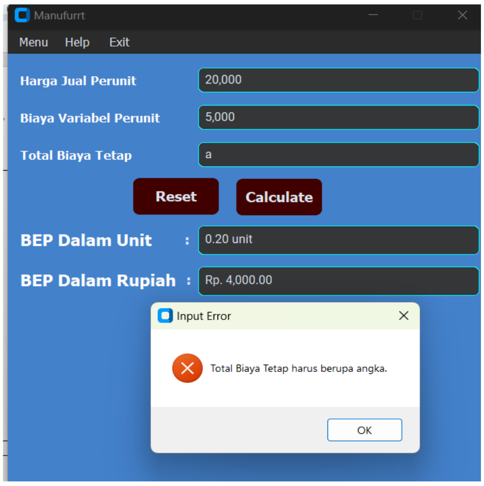

Figure 11. Error notification in the form a message box

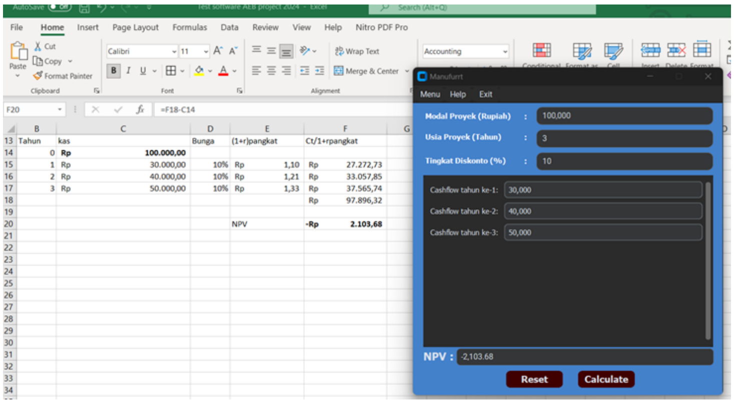

Figure 12. Expected result vs real result on NPV first scenario

Despite achieving a high average user acceptance score of 90% in the User Acceptance Test (UAT), minor resistance was initially observed, particularly from users who were less familiar with Python-based desktop environments. This phenomenon aligns with findings in digital tool adoption literature, which suggest that unfamiliarity with underlying technologies can hinder early user engagement and trust in system outputs. To mitigate this, several strategies were implemented. First, the application was deliberately designed with minimalist and intuitive GUI, employing consistent button placements, clear input-output labels, and error handling prompts to reduce user disorientation. Second, structured user guidance in the form of short tutorial documents and scenario-based walkthroughs were provided during the testing phase. These materials enabled users to simulate real-world calculations, reinforcing understanding through experiential learning. Finally, feedback loops during UAT were used to iteratively refine usability aspects of the application, ensuring that even novice users could perform feasibility calculations with confidence. This approach effectively minimized initial cognitive load and facilitated user adaptation, contributing to the application's positive reception, and validating its readiness for broader implementation across diverse user profiles in manufacturing-related investment projects.

Implementation Case

After conducting UAT testing by 3 end-users, the next process is to apply the calculation process using the application and compare the results of use on real case data from 3 reference sources where each end-user proves the calculation process using Excel, Manuffurt app and corrects the calculation results from the data reference source for each type of industrial scope used as references.

Case A: Net Present Value Calculation for Hydraulic Press Machines Production Business

Based on [38], Astana Wira Karya Ltd, as one of manufacturing industry located in Purwokerto, Banyumas, which operates in the production of hydraulic press machines, conducted an economic feasibility

SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 2, June 2025, pp.365 - 377
ISSN 2407-0939 print/ISSN 2721-2041 online

study using both Microsoft Excel and the Manufurrt application. The analysis focused on the Net Present Value (NPV) indicator over a 4-year project period with a total investment of Rp. 208,700,000 and a discount rate of 6.1% per year, derived from the average national inflation rate. The annual projected net benefit was consistent at Rp. 318,400,000, and when discounted over the project's lifetime, the resulting NPV was calculated to be Rp. 892,067,955.21. Both the spreadsheet-based method (Excel) and the Manufurrt application produced the same output, as illustrated in Figure 13. This reinforces the validity of the calculation engine embedded within Manufurrt.

This equivalence not only strengthens confidence in the application's accuracy but also highlights the efficiency gained by the Manufurrt app automated the entire process, minimizing manual formula entry and eliminating the risk of human error, which is common in spreadsheet use. Thus, Manufurrt serves as a reliable decision-support tool for financial evaluation in manufacturing investment projects, especially for practitioners who require quick, repeatable, and accurate results.

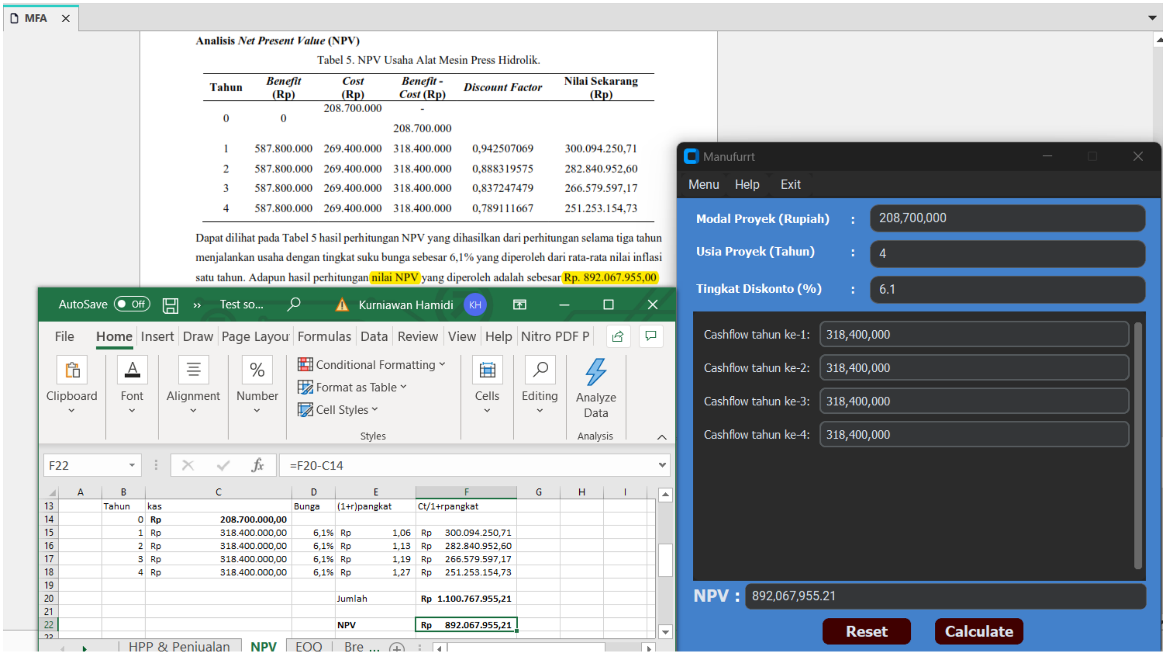

Figure 13. The results of NPV calculation with Manufurrt and Excel applications in the business case of the production of hydraulic press machines owned by Astana Karya Ltd (data source: [38])

Case B: Net Present Value Calculation for on investment feasibility analysis of rooftop solar power plants with on grid systems in ready-to-drink beverage factories

Indonesia is challenged with the issue of fulfilling the national energy policy. As a concrete step to support the policy, the ready-to-drink beverage factories is planning for the implementation of renewable energy by using rooftop solar power plants with an on-grid system to save expenses on electricity usage costs. Therefore, several feasibility analyses were carried out using several indicators, one of which was NPV. Based on data from [39] the beverage factory uses an investment capital of Rp. 3,520,000,000 and a discount factor of 5% for 25 years with data and calculation results in Figure 14. The results of calculations using spreadsheets and the Manuffurt application found that the plan to save energy by investing in rooftop solar power plants with an on-grid system for 25 years, resulted in a positive NPV value of Rp. 4,511,142,211.83.

Case C: Net Present Value Calculation for Feasibility Analysis of Heavy Equipment Investment at PLWJ Company

Based on [40], the PLWJ company, a natural resources mining firm specializing in quarrying sand and stone materials, aims to enhance its productivity and operational efficiency by investing in new heavy equipment. The current equipment fleet has become obsolete, leading to decreased performance and rising maintenance costs. As a result, a comprehensive financial feasibility analysis was deemed necessary to support informed decision-making regarding investment. This equivalence not only strengthens confidence in the application's accuracy but also highlights the efficiency gained by the Manufurrt app automated the entire process, minimizing manual formula entry and eliminating the risk of human error, which is common in spreadsheet use. Thus, Manufurrt serves as a reliable decision-support tool for financial evaluation in manufacturing investment projects, especially for practitioners who require quick, repeatable, and accurate results.

Table 6. Beverage factory investment data (source: [40])

Project Capital	Rp. 8.495.860.000
Project Age	3 years
Discount Rate (%)	9.95 %
Cash flow year 1	Rp. 23.860.510.000
Cash flow year 2	Rp. 20.366.220.000
Cash flow year 3	Rp. 13.499.530.000

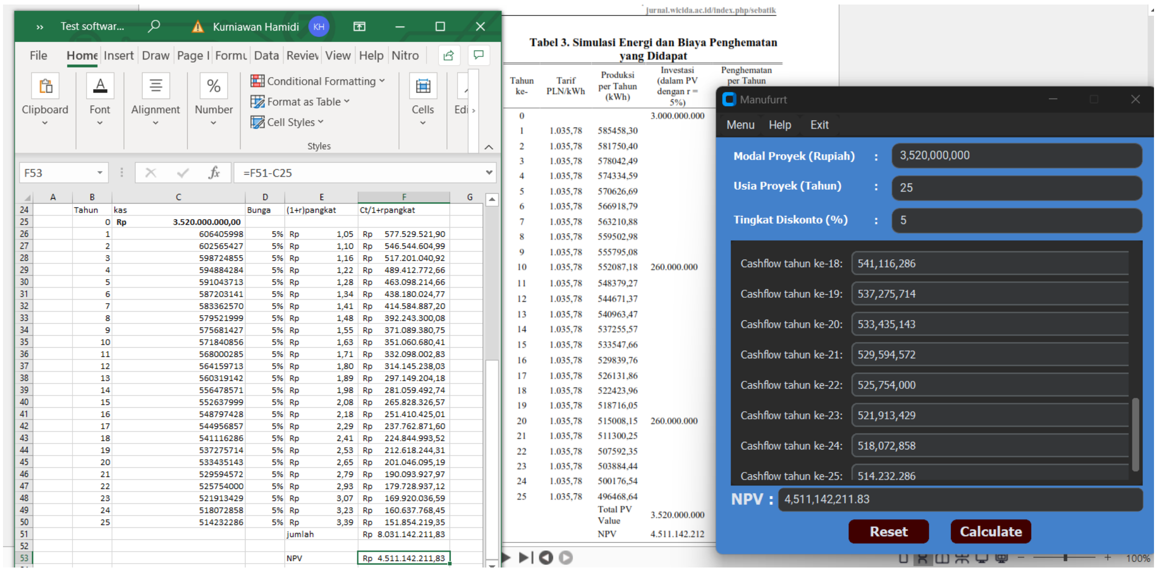

Figure 14. NPV calculation results with Manufurrt and Excel applications on investment feasibility analysis of rooftop solar power plants with on-grid systems in ready-to-drink beverage factories (data source: [39])

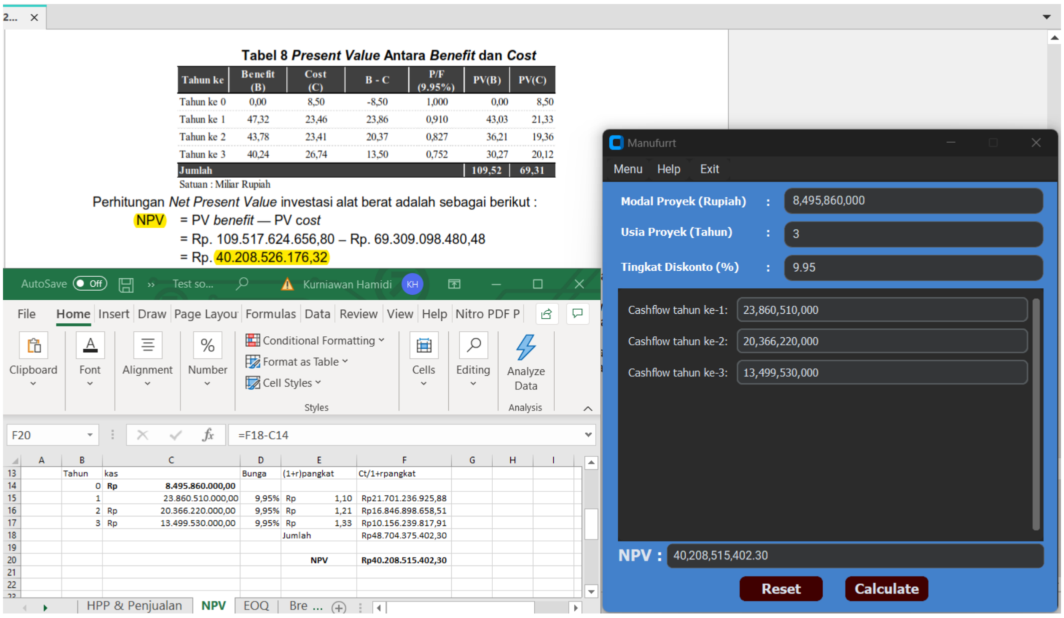

Figure 15. NPV calculation results with Manufurrt and Excel applications in the case of heavy equipment investment belonging to a sand and stone natural material quarry mining company (data source: [40])

Conclusion

Based on the results of the research that has been carried out, it can be concluded that the development of the Manufurrt application has succeeded in being an effective solution in overcoming the obstacles of the project feasibility test process. This application was developed using the SDLC waterfall method combined with generative AI in the design and coding completion phases. The UAT test results and implementation test results from secondary empirical data show that the Manufurrt application is interactive and feasible to use for investment and operational feasibility analysis.

The suggestions for further research in the future are the addition of data integration features to local and or cloud databases to store calculation histories and project feasibility test results reports and add multi-scenario analysis features so that they can compare analysis results in 1 view and add multiple method simulation features to obtain calculation results when there is uncertainty.

References

[1] M. Peña Abreu, C. R. Rodríguez Rodríguez, P. Y. Piñero Pérez, and Y. García García, "New Methods for Feasibility Analysis of Investment Projects in Uncertain Environments," in *Artificial Intelligence in Project Management and Making Decisions*, P. Y. Piñero Pérez, R. E. Bello Pérez, and J. Kacprzyk, Eds., Cham: Springer International Publishing, 2022, pp. 143–154. doi: 10.1007/978-3-030-97269-1_8.

[2] R. Abdulwahhab, S. Naimi, and R. Abdullah, "Managing Cost and Schedule Evaluation of a Construction Project via BIM Technology and Experts' Points of View," *Math. Model. Eng. Probl.*, vol. 9, no. 6, pp. 1515–1522, 2022, doi: 10.18280/MMEP.090611.

[3] M. Dombrowski, Z. Dombrowski, J. Woloszyn, A. Sachenko, O. Sachenko, and I. Melnychuk, "Adaptive Management of Digitalization Projects for Efficiency Increasing," in *Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021*, 2021, pp. 1195–1199. doi: 10.1109/IDAACS53288.2021.9660388.

[4] C. Markou, G. K. Koulinas, and A. P. Vavatsikos, "Project resources scheduling and leveling using Multi-Attribute Decision Models: Models implementation and case study," *Expert Syst. Appl.*, vol. 77, pp. 160–169, 2017, doi: 10.1016/j.eswa.2017.01.035.

[5] I.-M. Jiang, Y.-H. Liu, and S. Pakavaleetorn, "Optimal Sequential Investment Decision-Making with Jump Risk," *Asia-Pacific J. Oper. Res.*, vol. 39, no. 04, p. 2140035, Sep. 2021, doi: 10.1142/S0217595921400352.

[6] J. R. Bertini Junior, M. A. Funcia, A. A. S. Santos, and D. J. Schiozer, "A comparison of machine learning algorithms as surrogate model for net present value prediction from wells arrangement data," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8. doi: 10.1109/IJCNN.2019.8851708.

[7] T. N. Uryadova and M. G. Leshcheva, "Analytical Tools to Justify Investment Decisions," *Res. Econ. Financ. Probl.*, no. 03, pp. 1–10, 2022, doi: 10.31279/2782-6414-2022-3-7-1-10.

[8] F. Alkaraan and D. Northcott, "Strategic capital investment decision-making : A role for emergent analysis tools ? A study of practice in large UK manufacturing companies," vol. 38, pp. 149–173, 2006, doi: 10.1016/j.bar.2005.10.003.

[9] D. Tikhomirov and V. Plotnikov, "The minimisation of risks in project finance : approaches to financial modelling and structuring," vol. 05069, 2018.

[10] A. F. Shorikov, E. V Butsenko, and V. A. Tyulyukin, "Intelligent software system for optimizing adaptive control of investment projecting processes," *AIP Conf. Proc.*, vol. 2302, no. 1, p. 60014, Dec. 2020, doi: 10.1063/5.0033499.

[11] R. Tormosov, I. Chupryna, G. Ryzhakova, V. Pokolenko, D. Prykhodko, and A. Faizullin, "Establishment of the rational economic and analytical basis for projects in different sectors for their integration into the targeted diversified program for sustainable energy development," in *2021 IEEE International Conference on Smart Information Systems and Technologies (SIST)*, 2021, pp. 1–9. doi: 10.1109/SIST50301.2021.9465993.

[12] M. Bavay, M. Reisecker, T. Egger, and D. Korhammer, "Inishell 2 . 0 : semantically driven automatic GUI generation for scientific models," pp. 365–378, 2022.

[13] A. F. Vianiryzki and G. S. Niwanputri, "A User-Centered Approach to Design a Financial Asset Investment Mobile Application for Building Investing Eagerness," in *2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, 2021, pp. 1–6.

doi: 10.1109/ICAICTA53211.2021.9640286.

[14] M. Kolhekar, S. Upadhyay, H. Yeole, N. Patil, and V. Totawar, "Development of User-friendly Stock Prediction Assistance using Ensemble Learning," in *2024 Asia Pacific Conference on Innovation in Technology (APCIT)*, 2024, pp. 1–6. doi: 10.1109/APCIT62007.2024.10673583.

[15] W. Sardjono and A. Bhagas, "ACCELERATE DECISION-MAKING PROCESS THROUGH THE IMPLEMENTATION OF FINA: FINANCIAL FEASIBILITY STUDY IN HAND," *J. Theor. Appl. Inf. Technol.*, vol. 101, no. 23, pp. 7512–7519, 2023.

[16] S. Boyd, "REFeasibility: Designing a mobile application for initiating feasibility analysis," *Pacific Rim Prop. Res. J.*, vol. 21, no. 2, pp. 179–196, 2015, doi: 10.1080/14445921.2015.1058035.

[17] F. K. Foo and D. S. C. Ong, "Advance Injection Strategy Optimization: Maximize Benefit-Cost Ratio by Integration of Economic Spreadsheet in Excel to Assisted History Matching Using Python Scripting," in Abu Dhabi International Petroleum Exhibition and Conference, vol. Abu Dhabi. 2021, p. D032S237R002. doi: 10.2118/207955-MS.

[18] M. Moshiri, M. Raza, M. Sahlab, A. A. Malik, A. Bilberg, and G. Tosello, "Value Chain Comparison of Additively and Conventionally Manufactured Multi-Cavity Tool Steel Inserts: An Injection Molding Industrial Case Study for High-Volume Production," *Appl. Sci.*, vol. 12, no. 20, 2022, doi: 10.3390/app122010410.

[19] S. Pullteap, P. Samartkit, K. Kheovichai, and H. C. Seat, "A software development for investment analysis of LED lighting production project using fuzzy logic technique," vol. 14, pp. 83–100, 2020.

[20] A. S. Albana and Y. Andrian Saputra, "Financial Risk Assessment For Power Plant Investment Under Uncertainty Using Monte Carlo Simulation," in *2019 International Conference on Technologies and Policies in Electric Power & Energy*, 2019, pp. 1–6. doi: 10.1109/IEEECONF48524.2019.9102631.

[21] S. Mathews, "Valuing Risky Projects with Real Options," *Res. Manag.*, vol. 52, no. 5, pp. 32–41, 2009, doi: 10.1080/08956308.2009.11657587.

[22] D. LI, P. Chang, X. YANG, Z. SUN, F. JIN, and Y. YIN, "Two Dimensional Monte-Carlo Simulation Method of Risk Assessment for Strategic Asset Investment Decision Making," in SPE Annual Technical Conference and Exhibition, vol. SPE Annual. 2015, p. D021S021R004. doi: 10.2118/174885-MS.

[23] B. Losada, J.-M. López-Gil, and M. Urretavizcaya, "Improving Agile Software Development Methods by means of User Objectives: An End User Guided Acceptance Test-Driven Development Proposal," in *Proceedings of the XX International Conference on Human Computer Interaction*, in Interacción '19. New York, NY, USA: Association for Computing Machinery, 2019. doi: 10.1145/3335595.3335650.

[24] K. C. Bourne, *Application Administrators Handbook: Installing, Updating and Troubleshooting Software*. USA: Elsevier Inc., 2013.

[25] M. A. Dhina, G. Hadisoebroto, I. E. Setiawati, G. Undang, M. Masluh, and S. R. Mubaroq, "Digital performance assessment: Measure a pharmacy physics laboratory's skill," *J. Phys. Conf. Ser.*, vol. 1869, no. 1, pp. 0–6, 2021, doi: 10.1088/1742-6596/1869/1/012099.

[26] L. Nägele *et al.*, "A backward-oriented approach for offline programming of complex manufacturing tasks," in *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, 2015, pp. 124–130. doi: 10.1109/ICARA.2015.7081135.

[27] A. Siddieq and I. Nurhaida, "Mobile application of BTS tower search build upon location based service (LBS)," *Libr. Hi Tech News*, vol. 36, no. 3, pp. 1–6, Jan. 2019, doi: 10.1108/LHTN-08-2018-0049.

[28] N. A. Khan, "Research on Various Software Development Lifecycle Models," in *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 3*, K. Arai, S. Kapoor, and R. Bhatia, Eds., Cham: Springer International Publishing, 2021, pp. 357–364.

[29] T. Brinck, D. Gergle, and S. D. Wood, "CHAPTER 2 - User Needs Analysis," in *User Experience Re-Mastered*, C. Wilson, Ed., Boston: Morgan Kaufmann, 2010, pp. 23–72. doi: https://doi.org/10.1016/B978-0-12-375114-0.00005-0.

[30] Q. Liu *et al.*, "Python-Based Data Analysis Tool for a 6-DoF Industrial Robot Python-Based Data Analysis Tool for a 6-DoF Industrial Robot," 2019, doi: 10.1088/1757-899X/568/1/012098.

[31] M. Suhaib, "Conflicts identification among stakeholders in goal oriented requirements engineering process," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 12, pp. 4926–4930, 2019, doi: 10.35940/ijitee.L3557.1081219.

[32] K. Hamidi, L. Hanarisanty, R. I. Kurniati, and H. Wilujeng, "Rancang Bangun Aplikasi Dekstop untuk Kebutuhan Uji Fitoremediasi," vol. 24, pp. 208–216, 2024.

[33] J. H. Liu, H. X. Xia, and H. B. Zhang, "A Research into the UML Legend in the Waterfall Model Development," in *Computer and Information Technology*, in Applied Mechanics and Materials, vol.

519. Trans Tech Publications Ltd, 2014, pp. 322–328. doi: 10.4028/www.scientific.net/AMM.519-520.322.

[34] Z. Y. Oguz and C. Turhan, "Ontology-Based Semantic Analysis of Software Requirements: A Systematic Mapping," in *3rd International Informatics and Software Engineering Conference, IISEC 2022*, 2022. doi: 10.1109/IISEC56263.2022.9998243.

[35] A. ALazzawi, Q. M. Yas, and B. Rahmatullah, "A Comprehensive Review of Software Development Life Cycle methodologies: Pros, Cons, and Future Directions," *Iraqi J. Comput. Sci. Math.*, vol. 4, no. 4, pp. 173–190, 2023, doi: 10.52866/ijcsm.2023.04.04.014.

[36] M. Fischer and C. Lanquillon, "Evaluation of Generative AI-Assisted Software Design and Engineering: A User-Centered Approach," in *Artificial Intelligence in HCI*, H. Degen and S. Ntoa, Eds., Cham: Springer Nature Switzerland, 2024, pp. 31–47.

[37] M. Alenezi and M. Akour, "AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions," *Appl. Sci.*, vol. 15, no. 3, pp. 1–26, 2025, doi: 10.3390/app15031344.

[38] M. F. Atthoriq *et al.*, "Analisis Kelayakan Finansial Usaha Produksi Mesin Press Hidrolik," *JATI*, vol. 02, no. 02, pp. 28–35, 2025, doi: 10.31962/jati.v2i2.308.

[39] M. F. Hiswandi, F. Iswahyudi, and W. M. Soeroto, "Analisis Kelayakan Investasi Pembangkit Listrik Tenaga Surya Atap Dengan Sistem on-Grid Di Pabrik Minuman Siap Saji," *Sebatik*, vol. 27, no. 1, pp. 22–29, 2023, doi: 10.46984/sebatik.v27i1.2246.

[40] F. S. Hanafi, K. L. Mandagie, and D. A. N. H. Moektiwibowo, "Analisis kelayakan investasi alat berat dengan metode NPV , IRR dan NET B / C di perusahaan PLWJ," *J. Tek. Ind.*, vol. 9, no. 2, pp. 136–148, 2020, [Online]. Available: https://journal.universitassuryadarma.ac.id/index.php/jtin/article/view/645

[41] S. S. Kumar and J. Shetty, "Malicious PE File Detection Using Machine Learning: An Analysis of Header Features," in *2024 International Conference on Computing, Semiconductor, Mechatronics, Intelligent Systems and Communications (COSMIC)*, 2024, pp. 66–71. doi: 10.1109/COSMIC63293.2024.10871898.

[42] J. R. Dora and L. Hluchy, "Exploitation of Thick Client Application Vulnerabilities and a Synopsis of Mitigation: *How to Conduct Attacks to Abuse Weaknesses Present in a Windows Executable File.," in *2024 IEEE 18th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 2024, pp. 431–436. doi: 10.1109/SACI60582.2024.10619849.