PASCAL Journal of Computer Science and Informatics
https://jurnal.devitara.or.id/index.php/komputer
E-ISSN: 3047-5074 Volume 2 Nomor 2 Tahun 2025

Development of a Mini Calculator Application for Processing Odd and Even Numbers Using Dev-C++

[1]Indah Rahma Sari, [2]Amelia Anggraini, [3]Asiawatie S, [4]Zet Abdullah, [5] Novita Sari, [6]Hijrah Hasanah Putri
[1,2,6] Department of Information Systems, Institut Teknologi dan Bisnis Bina Sriwijaya Palembang, Indonesia
[3, 4] Department of Psychology, Sekolah Tinggi Ilmu Psikologi Widya Darma Palembang, Indonesia
[5]Department of English, Institut Teknologi dan Bisnis (ITB) Bina Sriwijaya Palembang, Indonesia

ARTICLE INFO

Keywords:
Even Numbers,
Odd Numbers,
Calculator,
Mini,
Dev-C++

Email:
macesmad@gmail.com

ABSTRACT

A mini calculator is a simple application designed to perform basic arithmetic operations such as addition, subtraction, multiplication, and division. This mini calculator program was developed using Dev-C++ with the primary goal of providing a practical solution for users to perform numeric calculations interactively. During the development process, several key issues were identified, including difficulties in accessing the calculator after the initial calculation, input errors, and the program's inability to handle repeated operations without restarting the application. The development method utilized in this application was procedural programming, incorporating the use of the do-while loop structure. This approach enables the calculator to continuously request input and perform computations based on user choices without the need to close or restart the program. The do-while loop ensures that users can carry out multiple operations in succession until they choose to exit. To address the aforementioned issues, several solutions were implemented. These include input validation mechanisms to handle errors such as division by zero or invalid inputs, as well as an interactive menu system that facilitates operation selection. Moreover, error handling features were incorporated to provide users with clear messages in the event of incorrect operations. Separate functions were implemented for each mathematical operation to enhance code modularity and maintainability. The development stages of this application involved the following steps: (1) designing the algorithm and calculator menu structure; (2) implementing individual functions for each arithmetic operation; (3) applying the do-while loop to control the program's iterative flow; (4) conducting testing and debugging to ensure proper functionality; and (5) refining the user interface and error messages to enhance user experience. In conclusion, the use of the do-while loop as a program repetition method has proven to be effective in maintaining the calculator application's interactivity without requiring a restart. The program provides an efficient and user-friendly solution for repeated calculations through a simple and responsive interface.

Copyright © 2025 PASCAL.
All rights reserved is Licensed under a Creative Commons Attribution- NonCommercial 4.0 International License (CC BY-NC 4.0)

INTRODUCTION

The Introduction to Programming course at ITB Bina Sriwijaya is a foundational subject that explores basic C++ programming concepts, including algorithms, basic program structures, data types, input and output variables, operators, strings, conditional statements, arrays, and looping mechanisms. In this course, Dev-C++ is used as the primary Integrated Development Environment (IDE) for writing and compiling code. Dev-C++ is a TDM-GCC-based compiler distributed under the GNU General Public License, providing an efficient and integrated environment for developing C++ programs, thereby saving time during the coding process.

Students enrolled in this course are expected not only to comprehend theoretical concepts but also to implement them through practical applications. These implementations are realized by developing simple programs, such as calculators or cashier systems, written in C++. To effectively support the achievement of learning objectives, it is necessary to use creative, innovative, and interactive learning media that can enhance student motivation and engagement in learning programming. This is essential because programming requires students to understand the function and syntax of various programming structures in C++.

Learning the C++ programming language can be challenging due to its numerous features and complexity. Therefore, to improve conceptual understanding, it is important to incorporate alternative instructional media that support learning outcomes (Rahmani, Syafrudin, & Wulandari, 2022). According to

Hakky, Sulindawati, and Permana (2018), effective learning media should integrate computer-based information and communication technologies to support programming instruction.

C++ is a well-established and widely used language, particularly in education and software development. It is frequently employed in the development of system software, games, and other high-performance applications. For students pursuing technology-related disciplines, proficiency in C++ is crucial. Through both theoretical and practical learning experiences, students can improve their knowledge of programming while also learning the correct syntax and structure of coding. One example of a practical application in this context is the development of a mini calculator program using Dev-C++.

A mini calculator developed in Dev-C++ typically refers to a basic application designed to perform standard arithmetic operations such as addition, subtraction, multiplication, and division. It is created using the C++ language and executed within the Dev-C++ IDE. The goal of this application is to simplify the calculation process for users via a command-line interface.

In previous projects, the calculator program utilized while loops and if-else statements to guide user interactions. However, this approach proved ineffective, as users were required to restart the application after each calculation. Additionally, requiring users to repeatedly select operations after every computation diminished the user experience. Therefore, a more effective and efficient solution is necessary.

To address this issue, this study proposes the development of a mini calculator using the do-while loop structure and a prototype-based development methodology. The use of do-while allows the application to remain interactive and continuously operational without restarting, thereby improving functionality, usability, and user satisfaction.

METHODS

The method applied in developing this program is the Prototype model. According to Pressman (2012), system design and development can be approached using the prototype methodology. A prototype is not a complete or final system but a working model that can be evaluated and modified repeatedly until it meets user requirements. This method is particularly suitable for applications where user feedback is essential to achieving a functional and user-friendly system.

The prototype model consists of five main phases in the “Design and Development of a Mini Calculator Application”:

1. Communication – This stage involves initial communication and data collection aimed at analyzing user needs. Developers gather information regarding what users expect from the system. Before conducting development, assumptions are formed based on the applied theoretical framework. These assumptions, or hypotheses, are then tested empirically through data collection and in-depth analysis.
2. Quick Plan – This is the early planning stage where initial requirements are outlined. It includes system analysis focusing on technology and user needs. This phase determines the necessary input specifications, the expected output, and the processing logic required to convert input into meaningful output.
3. Modelling Quick Design – This phase involves designing the application workflow and interaction processes using modeling tools. In this study, the Unified Modeling Language (UML) was used to visualize how the actors and system components would interact. A Use Case Diagram is employed to represent the expected functionality and user interactions. It shows how users engage with the system and outlines system behavior in response to user actions.
4. Prototype Construction – During this phase, the system prototype is developed, tested, and refined. The mini calculator is created using the C++ programming language in the Dev-C++ IDE. Initial prototypes are tested for functionality, and necessary revisions are made based on test results and user feedback.
5. Deployment, Delivery, and Feedback – The prototype is evaluated and reviewed by users. Feedback from this stage is used to refine the prototype further, leading to the development of the final system version. The finalized product is then fully implemented and made ready for actual use.

PASCAL Journal of Computer Science and Informatics
https://jurnal.devitara.or.id/index.php/komputer
E-ISSN: 3047-5074 Volume 2 Nomor 2 Tahun 2025

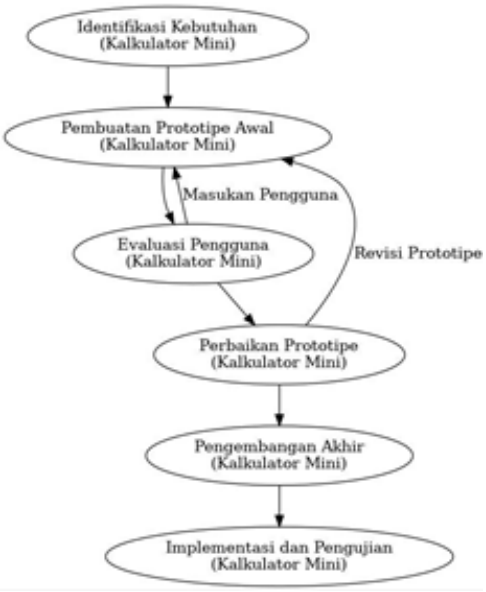

Figure 1. Steps in the Prototype Methodology

Implementation of the Do-While Loop in the Mini Calculator

One of the technical aspects in the mini calculator application involves choosing the right type of looping structure. In earlier implementations, while loops were used, which require the condition to be checked before entering the loop body. However, this caused limitations in user experience, especially when the loop conditions failed initially, resulting in skipped iterations.

The current application uses a do-while loop, where the condition is evaluated at the end of each loop cycle. This ensures that the loop body is executed at least once, regardless of whether the condition is true or false at the start. The do-while structure allows the program to continuously prompt users for input and perform operations until the user decides to exit, thus improving the interactivity and efficiency of the application.

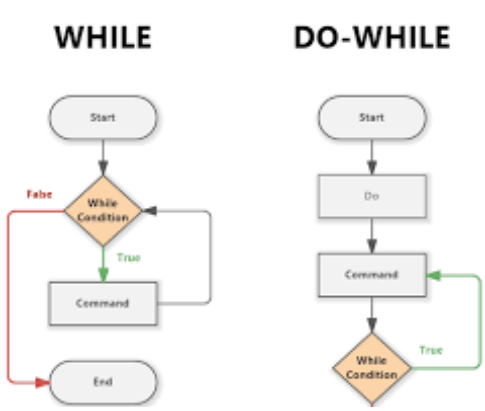

Figure 2. Do-While Loop Design for Mini Calculator

This loop structure is more suited for programs with uncertain iteration counts and enhances user experience by allowing repeated operations without restarting the application. It reflects a practical implementation of the prototype's interactive features and supports continuous testing and improvement.

RESULTS AND DISCUSSION

Following the completion of the design phase, the conceptual model was translated into a functional code to realize the proposed system. Based on the analysis and design stages, the results of this study indicate

that the mini calculator application effectively processes both odd and even numbers using an efficient looping structure. The implementation demonstrates the success of using iterative logic to support repeated user interactions without restarting the program.

The development of the mini calculator system involved the following operational steps:

1. Accessing the Dev-C++ IDE and implementing the pre-designed algorithm and code structure.
2. Executing the program by compiling and running the code.
3. Upon execution, the mini calculator interface prompts the user to input values (e.g., numbers and operators), and the system displays the calculated result.
4. If the user wishes to perform another calculation, they can choose to continue, and the program will repeat the operation as desired.

This mini calculator application was developed using the Prototype Methodology, which consists of the following stages:

1. Communication: This stage involved intensive communication between the developer and end users to understand the functional requirements and desired behavior of the mini calculator.
2. Quick Plan: This phase focused on drafting a preliminary development plan, identifying key functionalities, and defining system specifications.
3. Modeling Quick Design: Initial design concepts were modeled using flowcharts and diagrams outlining the system logic and operational techniques.
4. Use Case Diagram: A use case diagram was developed to visualize the interaction between users and the mini calculator system, depicting expected system behavior in response to user inputs.
5. Testing: The final stage included testing the implemented code to ensure proper functionality and to validate whether the system met the initial requirements.

The mini calculator program was written in the C++ language using the Dev-C++ IDE. The code integrates conditional logic and looping structures, specifically the do-while loop, which allows the application to continue processing calculations until the user decides to exit.

```
#include <iostream>
using namespace std;
int main() {
  char op;
  float num1, num2;
cout << "Enter operator: +, -, *, /: ";
  cin >> op;
cout << "Enter two operands: ";
  cin >> num1 >> num2;
  switch(op) {
    case '+':
cout << num1 << " + " << num2 << " = " << num1 + num2;
    break;
    case '-':
cout << num1 << " - " << num2 << " = " << num1 - num2;
    break;
    case '*':
cout << num1 << " * " << num2 << " = " << num1 * num2;
    break;
    case '/':
cout << num1 << " / " << num2 << " = " << num1 / num2;
    break;
    default:
// If the operator is other than +, -, * or /, error message is shown
cout << "Error! operator is not correct";
    break;
  }
return 0;
}
```

Figure 1. Mini Calculator Program Coding

PASCAL Journal of Computer Science and Informatics
https://jurnal.devitara.or.id/index.php/komputer
E-ISSN: 3047-5074 Volume 2 Nomor 2 Tahun 2025

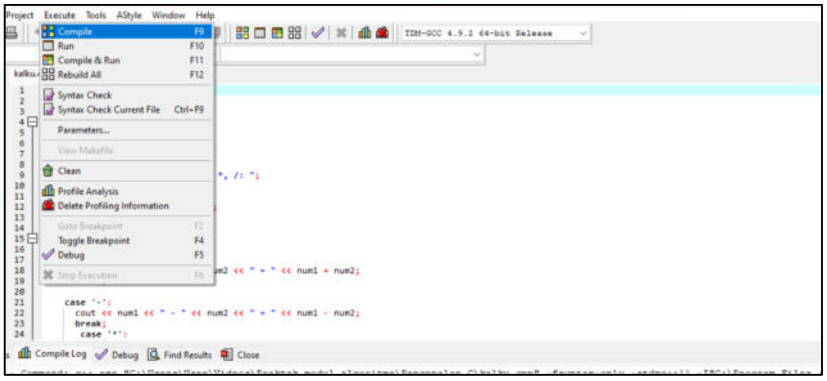

Figure 2. Program Compilation

This figure shows the Dev-C++ environment where the code is compiled to detect any errors before execution.

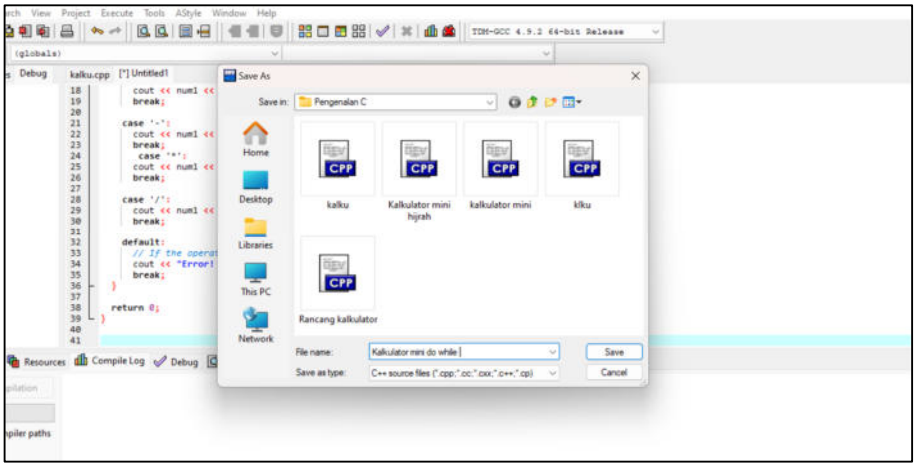

Figure 3. Saving the Program

Before running the program, it is saved to ensure the source file remains intact and reusable for future modifications.

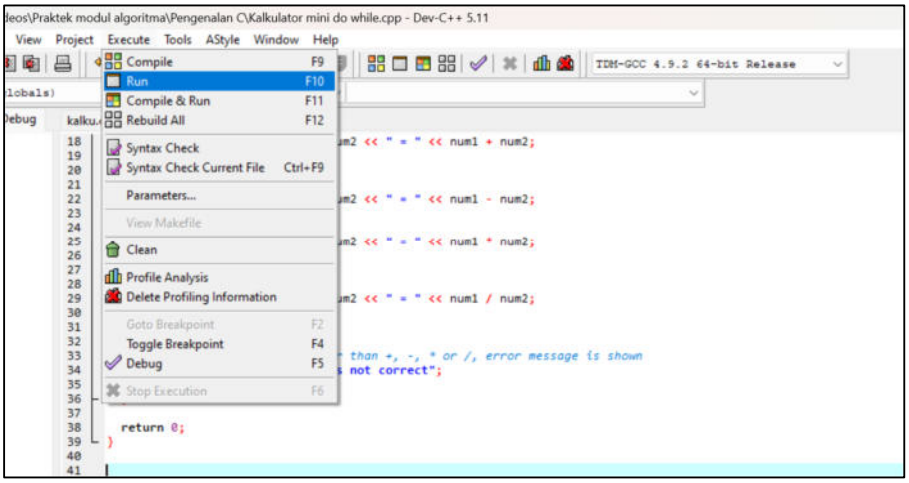

Figure 4. Program Execution

Upon execution, the system displays the user interface of the mini calculator, allowing users to input numbers and select arithmetic operations.

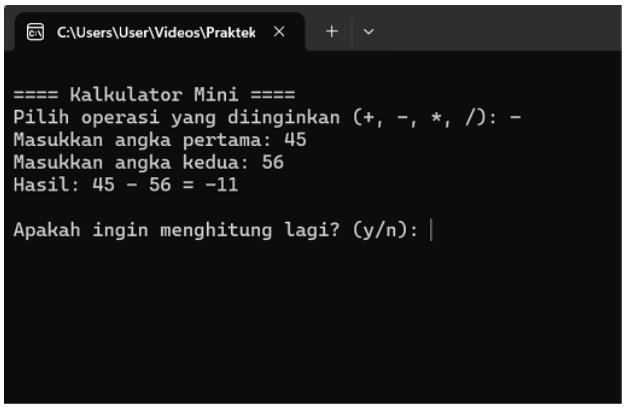

Figure 5. Output Display and Looping Interaction

In this final output view, the program displays the result of the arithmetic operation. It then prompts the user with the question: "Do you want to calculate again?", demonstrating the implementation of the do-while loop. This structure ensures that the user can perform multiple calculations in a single session without restarting the program.

This successful implementation validates the use of the do-while loop as an efficient method for maintaining continuous user interaction in a text-based calculator application. It enhances usability by reducing repetitive steps and supports the overall goal of making programming concepts more tangible for students learning C++.

CONCLUSION

Based on the research and development of the mini calculator application, it can be concluded that the system successfully classifies numbers and performs basic arithmetic operations accurately. The implementation using Dev-C++ demonstrates the flexibility of the language in solving simple computational problems. This study contributes to the understanding of fundamental programming concepts through the development of a simple yet functional application, which can serve as a foundation for more complex applications, particularly in the field of numerical computation. Through this project, the author has gained a deeper understanding of essential programming concepts such as variables, operators, conditional statements, and looping structures, as well as improved proficiency in using the Dev-C++ integrated development environment (IDE). It is recommended that future developments expand the functionality of this calculator to include more advanced operations, implement graphical user interfaces (GUI), and integrate error-handling mechanisms more extensively to further enhance user experience and system robustness.

REFERENCES

Aditya, R., Pranatawijaya, V. H., & Putra, P. B. A. A. (2021). Rancang bangun aplikasi monitoring kegiatan menggunakan metode prototype. Journal of Information Technology and Computer Science, 1(1), 47–57.

Fahriani, N. (2022). MENGENAL FUNDAMENTAL PROGRAMMING DENGAN DEVC++.

Hakky, M. M., Sulindawati, S., & Permana, R. (2018). Development of interactive learning media using information and communication technology in programming learning. Jurnal Pendidikan Teknologi dan Kejuruan, 15(2), 123–132.

Indahyanti, U., & Rahmawati, Y. (2020). Buku ajar algoritma dan pemrograman dalam bahasa C++ (pp. 1–146). Umsida Press.

Nita, S. (n.d.). Algoritma dan pemrograman-I. Deepublish.

Pradipta, A. A., Prasetyo, Y. A., & Ambarsari, N. (2015). Pengembangan web e-commerce Bojana Sari menggunakan metode prototype. eProceedings of Engineering, 2(1).

Pressman, R. S. (2012). Software engineering: A practitioner's approach (7th ed.). McGraw-Hill.

Putra, A. B. W. (n.d.). Pemrograman dasar pada program vokasi: Buku ajar. Media Nusa Creative (MNC Publishing).

Rahmani, A., Syafrudin, A., & Wulandari, I. (2022). The effectiveness of instructional media in improving programming skills among undergraduate students. International Journal of Computer Science and Education, 10(1), 45–52.

Taufiq, R. (2022). Algoritma dan pemrograman.

Uska, M. Z., Wirasasmita, R. H., & Kholisho, Y. N. (2023). E-modul dasar pemrograman berbasis Android sebagai media pembelajaran bahasa C++. Educate: Jurnal Teknologi Pendidikan, 8(2), 129–138.

Wali, M., Nengsih, T. A., Hts, D. I. G., Choirina, P., Awaludin, A. A. R., Yusuf, M., & Baradja, A. (2023). Pengantar 15 bahasa pemrograman terbaik di masa depan (referensi & coding untuk pemula). PT. Sonpedia Publishing Indonesia.