Automatic Java Code Generation System from Flowchart for Basic Programming Learning

Mustika Mentari[1], Pramana Yoga Saputra[2], Yan Watequlis Syaifudin[3], Imam Fahrur Rozi[4], Naufal Nafidiin[5]

[1, 2, 3, 4, 5] Information Technology Department, State Polytechnic of Malang, Indonesia
must.mentari@polinema.ac.id

Abstract. Basic Programming teachers invest considerable time and effort in preparing for their lessons, including creating questions, designing flowcharts, and crafting program code based on these questions and logical structures. While it's essential to maintain the necessary preparation, there is an opportunity to streamline the process. This study introduces a solution to simplify the conversion of flowcharts into Java program code through an automated code generator accessible via a web browser. This tool comprises a variety of symbols required for constructing flowcharts, enabling users to effortlessly translate their flowcharts into executable source code in the Java programming language. Users can then choose to either copy the generated Java code or save it for integration into other development environments. Furthermore, the system allows users to save their flowchart files for future reference. The development of this tool leveraged technologies such as JavaScript, D3graphvis, and the Laravel framework for creating an interactive user interface. Additionally, PhpMyAdmin MySQL was utilized for efficient database management. The System Usability Testing, which included ten core questions pertaining to the system, yielded promising results, with users providing positive feedback and an average System Usability Scale score of 78.5.

Keywords: Java Code, Generator, Flowchart, Basic Programming.

1 Introduction

Basic programming lecturers prepare many things that drain time and mind to teach programming, starting from making questions, flowcharts, and program code from questions and logic flows [1] [2]. Learning basic programming aims to provide an understanding of how to make computer programs. Students who will study basic programming in the future can work as programmers or in other information technology-related fields. The primary challenge faced in a computer programming course revolves around grasping the syntax and logic inherent in programming languages. Syntax, in this context, refers to the set of rules governing how a programming language's words and punctuation must be used. Essentially, every programming language comes with its own set of regulations regarding the correct usage of words and punctuation marks. Consequently, students frequently find themselves allocating a

R. Andrie Asmara et al. (eds.), *Proceedings of the 5th Annual Advanced Technology, Applied Science, and Engineering Conference (ATASEC) 2023*, Advances in Engineering Research 229,
https://doi.org/10.2991/978-94-6463-358-0_11

significant portion of their time to grappling with the intricacies of syntax, often at the expense of delving into the fundamental concepts of object orientation or problem-solving. Additionally, A programmer must develop the correct logic for a program to work properly nature of most programming environments works against the learning style of most students [3]. However, a program without syntax errors might still produce an incorrect result. A programmer must develop the correct logic for a program to work properly [4].

Tasks that many lecturers need to trim down without reducing the things that must be prepared before lectures begin. One of the stages that can be trimmed in the preparation of lecturers to teach basic programming is making answers in the form of program code, especially in practicum sessions. Making program codes as answers to practicum questions requires quite a long time and effort to think, which makes the brain rack up. Only with the condition that the flowchart is available can the logic of thinking for conversion in the program code be done automatically.

The training of educators to convert flowcharts into Java program code can be streamlined by implementing an automatic code generator [5]. This research suggests creating a tool that serves as an automatic code generator through a web browser interface, utilizing flowcharts for this purpose.

Visual formalisms have attracted great interest and have been heavily used in Computer Science and related fields. A characteristic example is Harel's state charts which resulted from an effort to define diagrams that describe the behavior of reactive systems [6].

Flowcharts have historically played a significant role in introductory Computer Science courses, serving as a valuable tool for acquainting newcomers with algorithms and programming [7]. Research literature indicates that these visual representations can be particularly effective for individuals who learn best through visual means, aiding both the creation and comprehension of algorithms [8].

This study proposes the development of an automated code generation tool, accessible through a web browser, which utilizes a well-structured flowchart as its foundation. This tool encompasses a variety of symbols necessary for constructing a comprehensive flowchart. Users can seamlessly convert their flowcharts into accurate Java programming language source code. A notable feature of this tool is its built-in debugging support, which facilitates the process of identifying and rectifying errors within a computer program [9]. Moreover, users can either copy or save the generated Java code for use in alternative integrated development environments (IDEs). Additionally, the system allows for the preservation of flowchart files for future reference.

This tool comprises a variety of symbols essential for crafting flowcharts. Users have the capability to transform these flowcharts into tangible source code composed in the Java programming language.

The resulting Java code can be copied or saved in different integrated development environments. Furthermore, the system is equipped to store flowchart files for future reference. The development of this tool employed JavaScript, D3graphvis, and the Laravel framework for constructing interactive user interfaces, along with PHPMyAdmin MySQL for effective database management [10].

The evaluation of System Usability Testing Tools, which included ten key questions regarding the system, yielded encouraging outcomes, with users providing favorable feedback. This resulted in an average SUS score of 78.5.

This paper consists of 3 sections: the first section discusses the introduction, the second section discusses related works, the third section describes the basic programming topics used in research in this paper, the fourth section discusses the core method regarding the Java code generator from flowcharts, section fifth discusses system implementation, section six discusses system evaluation, and section seven discusses conclusions.

2 Related Works

2.1 Code Generator

A design pattern-oriented approach was implemented to realise a state machine that incorporates hierarchical, concurrent, and historical states, all aimed at enabling Automatic Code Generation from UML State Chart Diagrams [11]. A new framework for Java code generation from a flowchart is also produced and verified for correctness [12]. Another reality is presented as a detailed overview of UJECTOR [13]. Implementation A tool inputs UML class, sequence, and activity to generate executable Java code. Combination activity diagram with sequence diagram as an input to generate code [14]. However, [15] presents the RM2PT tool based on their proposed approach to automated prototype generation from requirements models for requirements validation. The four case studies (LibMS, ATM, CoCoME and LoanPS systems) have been investigated, and the experiment result is satisfactory. There is also available proposes a tool and complete method to draw and convert flowcharts into C++ language [16].

2.2 Visual Programming

A program is symbolised through a diagram resembling a flowchart within a visual programming language. Students acquire programming skills by actively creating these diagrams. To employ this method, specialized integrated development environments (IDEs) are necessary. This visual technique is commonly employed to instruct beginners in programming, including schoolchildren, undergraduates, and those without prior computer science experience. Visual tools simplify the comprehension of fundamental programming concepts [17].

Visual tools also make the courses more interactive. As a result, several such tools have been developed. In this section, we list a set of tools that allow the implementation of visual programming approaches [14].

The RAPTOR tool offers capabilities like arrays, object-oriented programming, modular support, file input/output, and debugging services. However, it does not include features for exception handling or thread creation. RAPTOR's main objective is to improve students' problem-solving skills, and its stable version from 2015 is accessible to users at no cost [17].

The educational programming language, Scratch, created in 2005, operates on a block-based system. Users work by dragging and dropping programming blocks from a selection and assembling them together in a manner reminiscent of a jigsaw puzzle.

In contrast, the Microsoft VPL Tool, initially launched in 2006 for robotic application development, organises programs as sequences of activities linked to accomplish specific tasks. This tool is equipped with capabilities for creating arrays, lists, if-else conditions, variables, customized activities, debugging, and multitasking support. However, it does not include features for object orientation, file input/output, or exception handling.

Blockly, an open-source VPL is like scratch. A user may create a program using Blockly provided web interface and translate it into JavaScript, PHP, Dart, XML, and Python [18].

The Visual Logic tool presents a graphical interface that covers fundamental programming concepts such as variables, arrays, pre-and-posttest loops, if-else conditions, procedures, debugging, and improved input/output capabilities for console and text files. However, its primary limitation lies in its dependence on the operating system, as it can only function on Windows OS.

On the other hand, the Flowgorithm tool is a graphical application that enables developers to write and execute programs using flowcharts. It offers a straightforward method for creating computer programs that can be executed directly and converted into various source languages like C#, C++, Python, Perl, Java, and more. Flowgorithm's features include array creation, modular support, a graphical variable window for monitoring, support for multiple languages, loops, adaptable expressions, recursion, and a user-friendly interactive output system.

3 Topic of Basic Programming Learning

Basic Programming is an introductory-level course that utilizes the Java language. The Fundamentals of Programming course aims to impart knowledge and comprehension of fundamental algorithmic principles and basic programming techniques. Students acquire the foundation to tackle logical problems using flowcharts throughout this course. The basic programming learning system covers five key topics: Variables, Data Types, Operators, Input-Output; Selections; Looping; Arrays; and Functions. Detailed information on each of these topics is provided within the system.

1) Variables, Data Types, Operators, and Input-Output

Variables are used in programming languages to store temporary values which can be reused later. Variables have a data type and a name. Data type indicates the type of value in that variable.

2) Selections

Selection is an instruction which is used to select one possibility from several conditions, true or false. The general syntax used in Java for selection is "if" and "else".

3) Looping

Repetition or iteration are commands to repeat one or more statements several times. Loop statements are used so that we don't have to write one/a set of statements repeatedly. That way, typing errors can be reduced. Loops have two types, namely, Definite loops and Indefinite loops.

4) Arrays

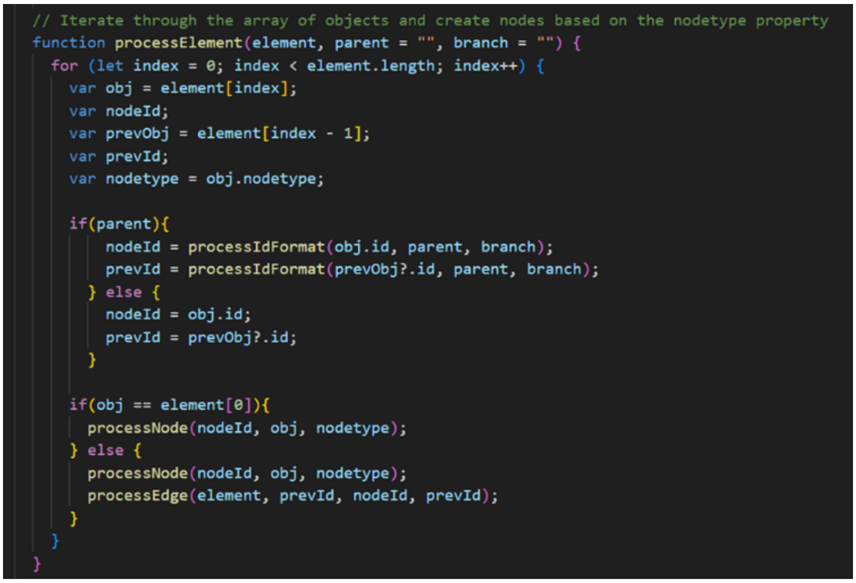
Arrays are complex variables with the same data type, use the same name, and have a certain index. It is a set of values (elements) with the same data type. Where each Array element can be accessed using a unique index

5) Function

A function is several instructions that are grouped together, stand-alone, which function to complete a particular job. If you use a function, the program can be arranged in a more structured (more modular) and effective way.

4 Java Code Generator from Flowchart

The user performs the program code generator by adding flowchart charts as needed, filling in the description label for each chart that has been selected, and determining the direction of the arrow indicating the continuation of the logical process.



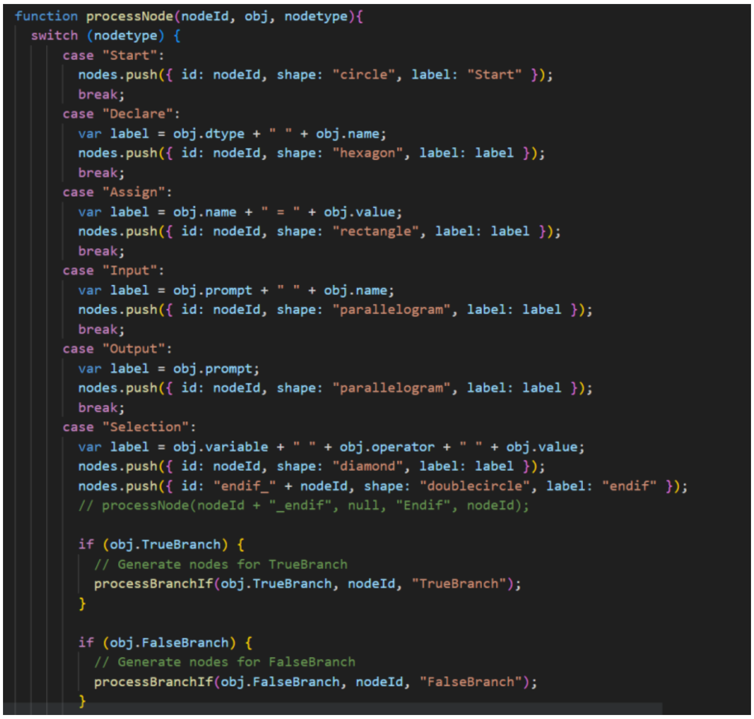
```
// Iterate through the array of objects and create nodes based on the nodetype property
function processElement(element, parent = "", branch = "") {
  for (let index = 0; index < element.length; index++) {
    var obj = element[index];
    var nodeId;
    var prevObj = element[index - 1];
    var prevId;
    var nodetype = obj.nodetype;

    if(parent){
        nodeId = processIdFormat(obj.id, parent, branch);
        prevId = processIdFormat(prevObj?.id, parent, branch);
      } else {
        nodeId = obj.id;
        prevId = prevObj?.id;
      }

    if(obj == element[0]){
      processNode(nodeId, obj, nodetype);
    } else {
      processNode(nodeId, obj, nodetype);
      processEdge(element, prevId, nodeId, prevId);
    }
  }
}
```

Fig. 1. ProcessElement Function

When the user opens the task page, a function-first will call existing JSON data in the database using Ajax. Initially, the JSON data will only consist of start nodes and end nodes. From there, the data will be manipulated to make a code and a graph based on the JSON data. It is also designed to enable users to interact with the graph by adding, editing, or deleting. This function is used to create dot language and render it into graphs. The function for receiving JSON data Ajax called in function refresh. It has a series of list functions to produce correct dot language.

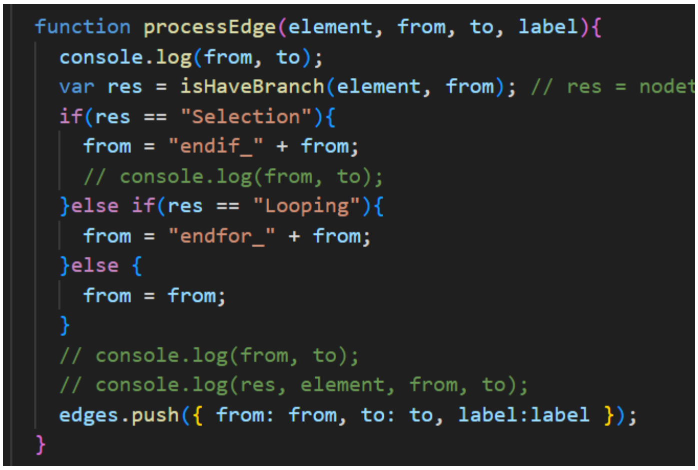


```
function processNode(nodeId, obj, nodetype){
  switch (nodetype) {
    case "Start":
      nodes.push({ id: nodeId, shape: "circle", label: "Start" });
      break;
    case "Declare":
      var label = obj.dtype + " " + obj.name;
      nodes.push({ id: nodeId, shape: "hexagon", label: label });
      break;
    case "Assign":
      var label = obj.name + " = " + obj.value;
      nodes.push({ id: nodeId, shape: "rectangle", label: label });
      break;
    case "Input":
      var label = obj.prompt + " " + obj.name;
      nodes.push({ id: nodeId, shape: "parallelogram", label: label });
      break;
    case "Output":
      var label = obj.prompt;
      nodes.push({ id: nodeId, shape: "parallelogram", label: label });
      break;
    case "Selection":
      var label = obj.variable + " " + obj.operator + " " + obj.value;
      nodes.push({ id: nodeId, shape: "diamond", label: label });
      nodes.push({ id: "endif_" + nodeId, shape: "doublecircle", label: "endif" });
      // processNode(nodeId + "_endif", null, "Endif", nodeId);

      if (obj.TrueBranch) {
        // Generate nodes for TrueBranch
        processBranchIf(obj.TrueBranch, nodeId, "TrueBranch");
      }

      if (obj.FalseBranch) {
        // Generate nodes for FalseBranch
        processBranchIf(obj.FalseBranch, nodeId, "FalseBranch");
      }
```

Fig. 2. processNode Function



```
function processEdge(element, from, to, label){
  console.log(from, to);
  var res = isHaveBranch(element, from); // res = nodet
  if(res == "Selection"){
    from = "endif_" + from;
    // console.log(from, to);
  }else if(res == "Looping"){
    from = "endfor_" + from;
  }else {
    from = from;
  }
  // console.log(from, to);
  // console.log(res, element, from, to);
  edges.push({ from: from, to: to, label:label });
}
```

Fig. 3. processEdge Function

Furthermore, processElement shown in Fig 1, is used to iterate the elements in the JSON data and perform node and edge creation using processNode shown in Fig 2 and processEdge functions, shown in Fig 3. The function which calls processNode is used to process the node contained in JSON data to transform it into a node in dot language, here, the nodetype of the data defines how data is translated into dot language it is including its id, shapes, and label. processEdge is used to define each node relation in accordance with the structure in the JSON data. Later, nodes and edges are collected in the format of dot language and rendered by d3. graphviz.

5 System Implementation

5.1 Architecture

The web-based system has been created utilising the Laravel framework. Laravel is an open-source PHP web framework that operates on the MVC (Model-View-Controller) architectural pattern. As depicted in Figure 4, the MVC framework facilitates interactions between the user and the database.

Within this MVC framework, the model component handles the data utilised in your application. It plays a crucial role in retrieving data from the database, performing necessary operations dictated by your application, and subsequently storing that data back into the database. The Model oversees the data exchanged between the database and the User Interface.

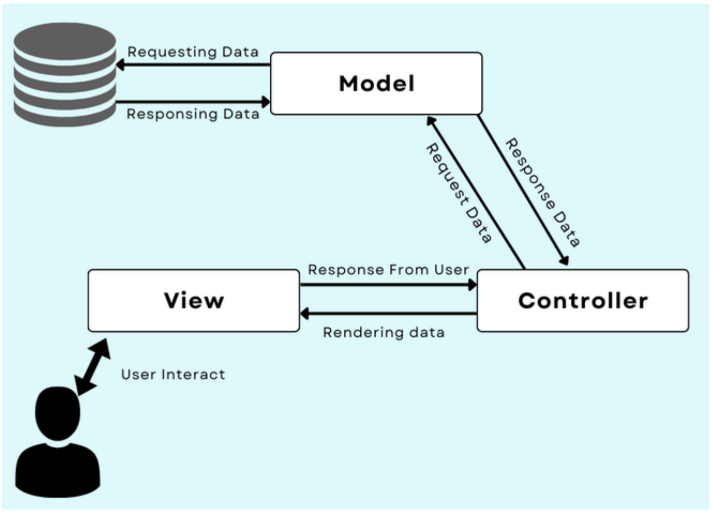


Fig. 4. Architecture System

The view component represents the User Interface, defining the template sent as a browser response. Within this component, the code presents data on the user's browser. For instance, it includes elements such as buttons, textboxes, dropdown menus, and various other widgets displayed on the browser screen.

On the other hand, the controller component plays a pivotal role in interacting with the model component. It retrieves data from the database and transfers it to the view component to generate the desired output on the user's browser screen. Similarly, when a user inputs data, the controller retrieves that information and then either performs specific operations or inserts it into the database through the model components.

The system architecture generally follows the visible MVC framework model. Java is used as a model (M), HTML/CSS/JavaScript as a framework for the View (V), and JSP for the controller (C).

5.2 Business Process

An overview of a business process can be seen in Fig 5. When the user opens the website, they are prompted to log in. After logging in, they can navigate to the task section. Initially, they will encounter a default-generated flowchart consisting of a start and end node. The interactive features allow users to add and edit nodes within the flowchart. The website offers various basic nodes for the flowchart, including declare, assign, input, output, decision, and loop nodes. The user can use the add and edit functions to draw the flowchart according to the given task.

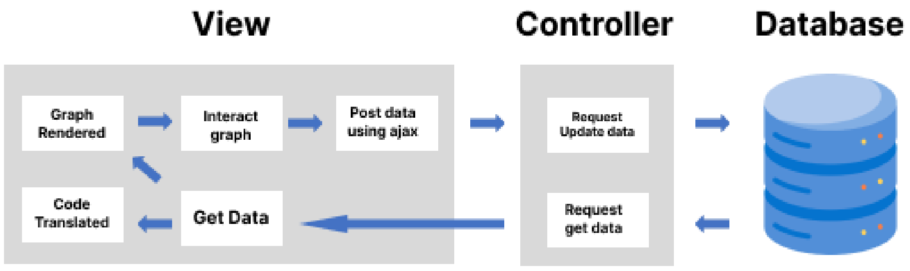


Fig. 5. Business Process Overview

Once the flowchart is completed, it is saved in a JSON format database. The flowchart is then translated into an array of code using a rule2code function. This code is further translated into HTML using the translate function. The JSON database is also transformed into an array of nodes and edges using the generate flowchart function. This Function consists of the processElement function, which traverses the JSON data performing processNode and processEdge to create nodes and edges to be collected and translated to dot language, later rendered using d3-graphviz, providing a visual representation of the flowchart to the user. Finally, the flowchart and corresponding code are instantly generated or updated after the user adds, edits, or deletes nodes.

5.3 Convert Question to JSON File

Before creating a flowchart, the user first creates questions in the form of story problems so that the program's logic flow can be mapped. If there are examples of case study questions like the one below, then the conversion results are in JSON form, as shown in Table 1.

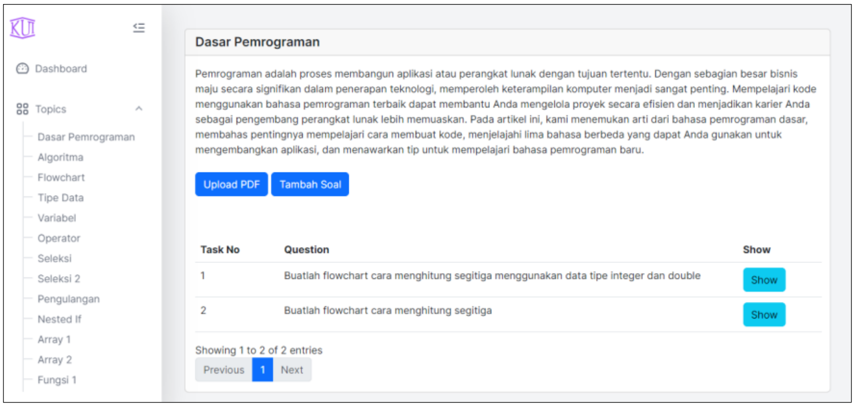
Question
"Sebuah sistem dibuat untuk menentukan pakaian dan peralatan yang harus dibawa pengguna sesuai dengan kondisi cuaca. Jika suhu lebih dari 27^{o} C, maka pengguna disarankan memakai dress, kemudian dilakukan pengecekan apakah saat ini hujan, jika hujan maka pengguna disarankan untuk membawa payung, sedangkan jika tidak hujan maka pengguna disarankan untuk memakai sunscreen. Namun, jika suhu kurang dari atau sama dengan 27^{o} C, maka pengguna disarankan memakai celana Panjang".

Table 1. The Result of translate to JSON Data

Json Data
[{"id":1,"nodetype":"Start"}, {"id":2,"nodetype":"Declare","name":"suhu","dtype":"int"}, {"id":3,"nodetype":"Declare","name":"hujan","dtype":"char"}, {"id":4,"nodetype":"Input","name":"suhu","prompt":"Masukkan"}, {"id":5,"nodetype":"Input","name":"hujan","prompt":"Masukkan"}, {"id":6,"variable":"suhu","operator":">","value":27,"nodetype":"Selection", "TrueBranch":[{"id":1,"nodetype":"Output","prompt":"Memakai Dress"},{"id":2,"variable":"hujan","operator":"equals","value":"y","nodetype":"Selection","Tru eBranch":[{"id":1,"nodetype":"Output","prompt":"Membawa payung"}],"FalseBranch":[{"id":1,"nodetype":"Output","prompt":"Memakai sun- screen"}]}],"FalseBranch":[{"id":1,"nodetype":"Output","prompt":"Memakai celana pan- jang"}]}, {"id":7,"nodetype":"End"}]

5.4 User Interface System

The main user interface display looks like Fig 6. Each topic contains a table of questions containing story questions, further showing the code generator page like Fig 7. The code generator page has two sides, left and right. The right side is where the flowchart is made, and the right side is part of the Java code conversion. In the flowchart creation section, the user can right-click on each shape to add a new shape and continue linking it to the next shape in the form of processing, input, selection and so on.



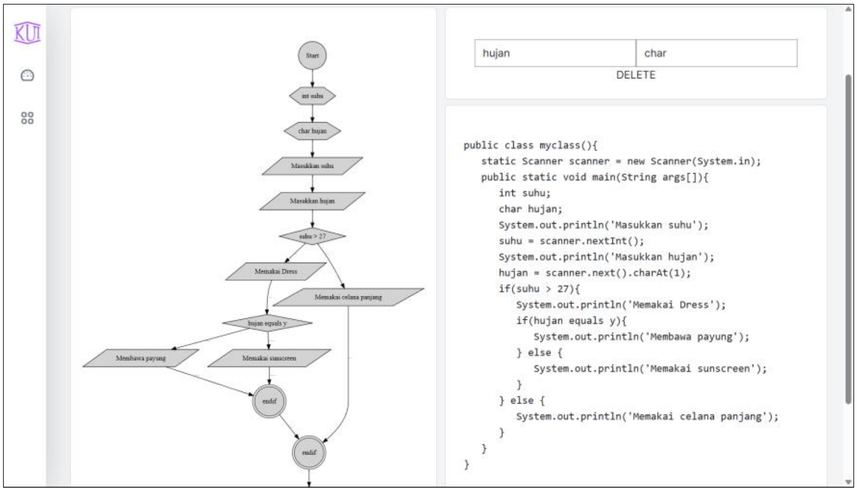
Dasar Pemrograman

Pemrograman adalah proses membangun aplikasi atau perangkat lunak dengan tujuan tertentu. Dengan sebagian besar bisnis maju secara signifikan dalam penerapan teknologi, memperoleh keterampilan komputer menjadi sangat penting. Mempelajari kode menggunakan bahasa pemrograman terbaik dapat membantu Anda mengelola proyek secara efisien dan menjadikan karier Anda sebagai pengembang perangkat lunak lebih memuaskan. Pada artikel ini, kami menemukan arti dari bahasa pemrograman dasar, membahas pentingnya mempelajari cara membuat kode, menjelajahi lima bahasa berbeda yang dapat Anda gunakan untuk mengembangkan aplikasi, dan menawarkan tip untuk mempelajari bahasa pemrograman baru.

Task No	Question	Show
1	Buatlah flowchart cara menghitung segitiga menggunakan data tipe integer dan double	Show
2	Buatlah flowchart cara menghitung segitiga	Show

Showing 1 to 2 of 2 entries

Fig. 6. Auto Generator Java Code User Interface



hujan	char

```
public class myclass(){
   static Scanner scanner = new Scanner(System.in);
   public static void main(String args[]){
      int suhu;
      char hujan;
      System.out.println('Masukkan suhu');
      suhu = scanner.nextInt();
      System.out.println('Masukkan hujan');
      hujan = scanner.next().charAt(1);
      if(suhu > 27){
         System.out.println('Memakai Dress');
         if(hujan equals y){
            System.out.println('Membawa payung');
         } else {
            System.out.println('Memakai sunscreen');
         }
      } else {
         System.out.println('Memakai celana panjang');
      }
   }
}
```

Fig. 7. Auto Generator Java Code User Interface Detil

6 System Evaluation

6.1 A Subsection Sample

System evaluation was applied to five lecturers teaching basic programming courses (participants). Testing the System Usability Scale System (SUS) uses ten types of questions about:

1) Desire to use the system again.
2) System usage requirements
3) ease of use of the system
4) The need for technical support when using the system.
5) The level of differentiating the system from the existing one

6) System inconsistency
7) System usage speed
8) Convenient system to use.
9) Easy-to-use system
10) Requires much preparation before using the system.

Table 2. Result of the SUS Test

Participant	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	Score
p1	3	4	4	2	5	1	4	2	5	2	75,0
p2	4	2	5	2	4	2	5	2	4	2	80,0
p3	4	3	4	3	4	1	5	2	5	2	77,5
p4	4	2	4	2	4	2	4	2	4	2	75,0
p5	5	2	4	2	5	2	5	1	4	2	85,0

The results obtained from the SUS test are shown in Table 2. The average value obtained is 78.5, with the highest value of 85 and the lowest value of 77. This shows the user's response to the Java code generator system is good.

7 Conclusion

This paper presents an auto generator Java code system from flowcharts for basic programming learning. The system is created by generating graphs from rule submissions stored in the database and stored in the JSON file. Using some of the functions that have been created, the graph will find out the initial and end nodes of each shape flowchart along with the identities of variable names, functions or other needs in the questions provided. Making a website-based system in this paper uses the Laravel framework, which includes View, Controller, and Database. The results of the SUS test on this system showed a good and positive response from users achieving an average SUS score of 78.5. This value shows a good response from the basic programming teachers when using the system code generator that has been made. In the future, a Java code generator website with more detailed topics is needed, not just topics in basic programming courses.

Acknowledgement

This endeavor would not have been possible without the cooperation of the entire research team and support from the State Polytechnic of Malang, especially the Information Technology Department.

References

1. Tsai, C.-Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. Computers in Human Behavior, 95, 224–232. https://doi.org/10.1016/j.chb.2018.11.038.
2. Gökoğlu, S., & Kilic, S. (2022). Programming learning and teaching of pre-service computer science teachers: Challenges, concerns, and solutions. E-Learning and Digital Media, 204275302211173. https://doi.org/10.1177/20427530221117331
3. V. Lian, E. Varoy, and N. Giacaman, "Learning Object-Oriented Programming Concepts Through Visual Analogies," *IEEE Transactions on Learning Technologies*, vol. 15, no. 1, pp. 78–92, Feb. 2022, doi: 10.1109/TLT.2022.3154805.
4. Y. Yang, X. Li, Z. Liu, and W. Ke, "RM2PT: A Tool for Automated Prototype Generation from Requirements Model," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, IEEE, May 2019, pp. 59–62. doi: 10.1109/ICSE-Companion.2019.00038.
5. Chen, H. (2020). Design and Implementation of Automatic Code Generation Method Based on Model Driven. *Journal of Physics: Conference Series*, *1634*(1), 012019. https://doi.org/10.1088/1742-6596/1634/1/012019
6. S. Xinogalos, "Using flowchart-based programming environments for simplifying programming and software engineering processes," in *2013 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, Mar. 2013, pp. 1313–1322. doi: 10.1109/EduCon.2013.6530276.
7. Ensmenger, N. (2016). The Multiple Meanings of a Flowchart. Information & Culture, 51(3), 321–351. https://doi.org/10.7560/IC51302
8. Denisov, M., Anikin, A., & Sychev, O. (2021). Dynamic Flowcharts for Enhancing Learners' Understanding of the Control Flow During Programming Learning (pp. 408–411). https://doi.org/10.1007/978-3-030-86062-2_42
9. Jobstmann, B., Staber, S., Griesmayer, A., & Bloem, R. (2012). Finding and fixing faults. Journal of Computer and System Sciences, 78(2), 441–460. https://doi.org/10.1016/j.jcss.2011.05.005
10. Susanto, A. (2019). Database Management System. INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH, 8(06). www.ijstr.org
11. E. V., S., & Samuel, P. (2019). Automatic Code Generation From UML State Chart Diagrams. IEEE Access, 7, 8591–8608. https://doi.org/10.1109/ACCESS.2018.2890791
12. C. Supaartagorn, "Web application for automatic code generator using a structured flowchart," in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, IEEE, Nov. 2017, pp. 114–117. doi: 10.1109/ICSESS.2017.8342876.
13. M. Usman, A. Nadeem, and T. Kim, "UJECTOR: A Tool for Executable Code Generation from UML Models," in *2008 Advanced Software Engineering and Its Applications*, IEEE, Dec. 2008, pp. 165–170. doi: 10.1109/ASEA.2008.39.
14. C. Redmon, K. Leung, Y. Wang, B. McMurray, A. Jongman, and J. A. Sereno, "Cross-linguistic perception of clearly spoken English tense and lax vowels based on auditory, visual, and auditory-visual information," *J Phon*, vol. 81, p. 100980, Jul. 2020, doi: 10.1016/j.wocn.2020.100980.
15. H. Yang *et al.*, "Multi-object tracking using Deep SORT and modified CenterNet in cotton seedling counting," *Comput Electron Agric*, vol. 202, p. 107339, Nov. 2022, doi: 10.1016/j.compag.2022.107339.

16. K. Charntaweekhun and S. Wangsiripitak, "Visual Programming using Flowchart," in *2006 International Symposium on Communications and Information Technologies*, IEEE, Oct. 2006, pp. 1062–1065. doi: 10.1109/ISCIT.2006.339940.
17. Kanika, S. Chakraverty, and P. Chakraborty, "Tools and Techniques for Teaching Computer Programming: A Review," *Journal of Educational Technology Systems*, vol. 49, no. 2, pp. 170–198, Dec. 2020, doi: 10.1177/0047239520926971.
18. B. Saha Tchinda, D. Tchiotsop, M. Noubom, V. Louis-Dorr, and D. Wolf, "Retinal blood vessels segmentation using classical edge detection filters and the neural network," *Inform Med Unlocked*, vol. 23, p. 100521, 2021, doi: 10.1016/j.imu.2021.100521.