

## DESIGNING AND QUALITY TESTING OF “DIGICHIP” VIRTUAL SIMULATION SOFTWARE OF ANDROID PLATFORM FOR MOBILE-VIRTUAL LEARNING SUPPORTING VOCATIONAL MECHATRONICS ENGINEERING

Titih Rejyasmito Hadi  
Universitas Negeri Yogyakarta

Eko Marpanaji  
Universitas Negeri Yogyakarta

### Abstract

*Technology enhancement in any life aspects is undeniably crucial to face the globalization era. A strategic approach to touch any life aspect is gained through education. Supporting this educational long-term goal, this study aims to: (1) produce learning media in the form of a virtual-learning simulation software, which is capable of operation on mobile devices to facilitate mobile, portable, effective, economical, and operationally-safe learning; (2) test the developed software quality in terms of the functional aspects of suitability, maintainability, portability, and usability (ISO/IEC 25010). The subjects of this Research and Development (R&D) study were class XI students of Mechatronics Engineering at SMTI Yogyakarta. The process of designing and testing software uses the V-model type Software Development Life Cycle (SDLC). Unit testing was done through the white-box technique with the base-path test, flowgraph, and independent path. The testing for integration, system, and acceptance used Black-box techniques. The study shows that: (1) V-models were used to design the "DigiChip" virtual simulation software to support virtual learning and mobile learning through the software development stage; (2) The software quality testing based on ISO/IEC 25010 shows that in the functional aspects suitability, all features function properly (very decent), its maintainability is of MI 84 (easy maintenance), it gains 100% portability, or can be operated on all Android OS kernels (very feasible), and its usability is 86.18% (very feasible) with Cronbach's Alpha 0.841 (good). Ninety percent of media experts and 100% of material experts consider that the developed software is very feasible.*

**Keywords:** simulation, virtual learning, mobile learning, mechatronics

**Permalink:** <http://dx.doi.org/10.21831/jpv.v9i2.23570>

**Contact** Titih Rejyasmito Hadi  [titih.rejyasmito@student.uny.ac.id](mailto:titih.rejyasmito@student.uny.ac.id)

 Department of Technology and Vocational Education,  
Graduate School of Universitas Negeri Yogyakarta  
Jl. Colombo No.1, Karangmalang, Depok, Sleman,  
Yogyakarta 55281, Indonesia

## INTRODUCTION

The development of science and technology is increasingly complex nowadays. Technology has an essential role in almost all aspects of human life needs. Similarly, the development of the world of digital technology and mobile computer technology has now been developed in the form of handhelds. The development of computer peripheral technology in an increasingly practical form makes various computer features can be enjoyed only on the hand. Technology is truly born to provide something useful and improve the quality of life for humankind. Technology is used as a learning tool and medium in the world of education, which will provide new productivity and skills. Thus, it must be supported by wise behavior in using computer handheld technology in a productive, educative direction, not even in the direction of being counterproductive. The wise behavior in using technology must be a top priority in giving birth to a generation that has a high level of quality in a nation.

Many people view handheld computer devices as the tools which can only be used as a means of entertainment and communication, even though the use of a more educative direction is very necessary to be published. One of them is simulation-based learning media and virtualization technology. Practical, efficient, economical, and safe levels, of course, become more points for simulation and virtual based learning media. It is expected to be very helpful in overcoming students' learning difficulties when faced with problems of limited tools, materials, time (study hours in class/lab/workshop), and learning systems that have to take turns and group.

A mobile device that is a Personal Computer (PC) device is certainly owned by each student personally. Learning activities in a personal and independent manner without having to depend on practical hours, tools, and lab materials provided by the school can be overcome. Students will have more hours of study and can study anytime, anywhere. This research is expected to provide education to the wider community and users of handheld computer devices as a medium that can be used in a more productive direction, in this case, as a medium for learning. Educators and students often use equipment such as; cellphone, smart-

phone, tablet-PC, which lately has become a device that is so familiar in everyday life. The development of computer technology up to now gave birth to new learning innovations based on this mobile device (Mobile Learning).

Learning activities using mobile devices can be used as an alternative in the problems of the digital engineering learning process, in the form of problems with the availability of equipment and lab materials. The main equipment is, of course, an I/O Digital Logic Board, which is an electronic hardware that functions to process the logic gate chip into an output in the form of a visual display that is usually in the form of an LED indicator light or in the form of segmentation. I/O Equipment Digital Logic Board is the main equipment of digital engineering practicum, which is difficult for each student individually. The availability of lab materials is also a problem in the process of digital engineering learning. The practicum material in the form of a logic gate chip is non-reusable, the arrays when damage occurs is difficult to repair. Damage to a logic gate chip can only be overcome by replacing a new logic gate chip. Because of the nature of the logic gate chip in its use when learning must operate on-off continuously in high intensity. Though when learning is possible, each student spends some practice material in the form of a logic gate chip.

Virtual simulation-based learning media will be more economical, efficient, portable, and safer to operate. Likewise, in this research development, the domain of mobile-based learning was chosen because mobile technology equipment in the form of smartphones and tablet-PCs was students' personal equipment, which the majority of them use and own. The effective value of learning that one student has one media learning will undoubtedly be realized. This development research designs and tests the quality of a learning medium that has the ability as a digital learning media in its original physical form, to be in the form of a virtual simulation. Another concept of innovation in simulation learning media products is the character of products that are off-line, so students are more comfortable to learn because they can use it without having to spend more on costs and depend on the internet connection. This development research, as its main purpose, is to provide innovation in the conversion of learning activities to digital techniques that

require high costs, become completely free of cost, as well as security and other operational conveniences. Research activities carried out to realize an adaptation and competitive nature of the development of the world of technology and vocational education are expected to be realized. Since the presence of the era of global competition requires the vocational education to develop and keep pace with the development in the world of work, therefore, it is important for vocational education to held competencies and remain relevant with demands of the world of business and industry in the era of global competition (Tamrin, Slamet, & Soenarto, 2018, p. 41).

## RESEARCH METHOD

This research used the Research and Development (R&D) research method. The software design and testing process used the V-model Software Development Life Cycle (SDLC) method, consisting of (Pressman, 2010, p. 40): (1) Requirement modeling, (2) Architectural design, (3) Component design, (4) Code generation, (5) Unit testing, (6) Integration testing, (7) System testing, and (8) Acceptance testing, as seen in Figure 1.

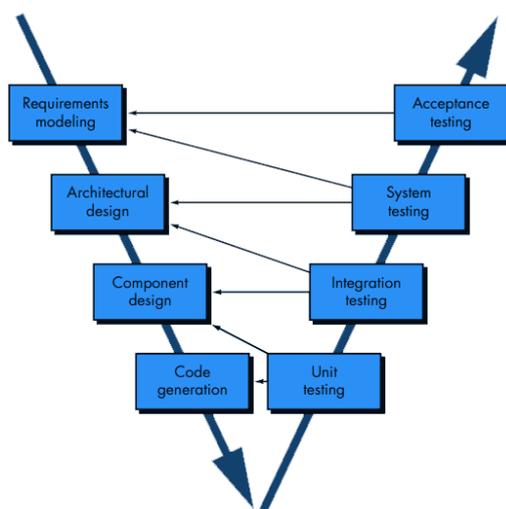


Figure 1. V-model Stage

Software quality testing refers to ISO 25010, in the functional aspects of suitability tested using a list of run tests and test cases. Maintainability aspects were tested through measurements of Maintainability Index (MI), Duplication Source Code, Line of Code (LoC), and Cyclomatic Complexity (CC). Portability

aspects were tested through installation activities on various types of hardware configurations and kernel versions of the Android OS. The usability aspect was tested using USE Questionnaire on users and Cronbach's Alpha calculations.

The Requirement modeling phase is carried out through a needs analysis activity in the form of potential user needs found in the school environment. Students are very limited by the problems of the limitations of tools, materials, time (study hours in class/lab), and learning systems that have to take turns or groups. The cost of the learning equipment is too high if students must have to study independently at home. The Architectural stage was done through analysis of media design concepts that are in accordance with digital engineering subject learning. Designing software such as what can overcome the problems faced by students, as well as what hardware can run the software made. Information was collected through observation and interviews.

The Component design stage was done by the process of designing components and systems according to the description of the system requirements analysis that was known in the previous stage. Component design is described in the Unified Modeling Language (UML), flowchart, and user interface (UI). The Code generation stage is an implementation program code design oriented to the system that has been made. The language of the programming code in this study is C#.

Phase Unit testing was done for the smallest part of the software, such as base-path, which consists of Flowgraph, Cyclomatic Complexity (CC), Independent Path to design a Test case. The CC calculations use elements in the form of edges and nodes (Najadat, Alsmadi, & Shboul, 2012, p. 2). Calculation of CC was conducted with the following formula (Equation 1) (McCabe, 1976, p. 308).

$$V(G) = e - n + p \quad (1)$$

Notes:

$V(G)$  = Cyclomatic Complexity

$e$  = number of edges on flowgraph

$n$  = number of node on flowgraph

$p$  = connected component

Connected Components ( $p$ ) has a constant value of the number "2" (Laplante, 2007,

p. 176). Unit testing is the first stage of testing carried out by the engineer or maker on the smallest part of the software made. The unit testing stage uses the white-box testing technique (Watkins & Mills, 2011, p. 52). Integration testing stage was done by testing the smallest part of the software that has been put together. Integration testing is carried out by the concerned engineer using the checklist method, run test, and test case. Integration testing was done to measure the level of functional suitability on ISO 25010 software quality parameters.

Stage System testing was carried out by the engineer concerned when all software components have become a whole unified system. System testing was done to measure the level of maintainability and portability of the software made. Maintainability aspects were tested through the measurements of Maintainability Index (MI), Code Duplication, Line of Code (LoC), Cyclomatic Complexity (CC). The portability aspect uses installation activities on various types of hardware configurations. The results of the duplication code calculation were converted to the rating table to find out the quality of the source code script. Table 1 presents the duplication assessment classification (Heitlager, Kuipers, & Visser, 2007, p. 7).

Table 1. Duplication Source Code Assessment Classification

	Rank	Duplication
++	very small	0-3%
+	small	3-5%
o	moderate	5-10%
-	big	10-20%
--	extremely big	20-100%

The acceptance testing stage was done through media experts, material experts, and users. The material experts are digital engineering subject teachers. The users or respondents are 59 students of Mechatronics Engineering Department of the Vocational High School-Industrial Engineering Vocational High School (or *Sekolah Menengah Kejuruan - Sekolah Menengah Teknik Industri (SMK-SMTI)*) class. The acceptance testing process was done to measure the usability aspects of the software by using a standard USE Questionnaire.

Testing on the functional aspects of suitability was done using a list of test cases that contain all the functions of the features in

the "DigiChip" software. The list of test cases refers to the format of Williams (2006, p. 44). Data analysis techniques in testing the functional aspects of suitability by giving a checklist to each test case that is able to function properly. The data of the test results are converted into percentages. The percentage obtained was then converted into definitions of feasibility. The feasibility category refers to the feasibility table proposed by Arikunto (2013, p. 35), as shown in Table 2.

Table 2. Feasibility Classification

Number (in %)	Classification
< 21	Very Not Feasible
21 - 40	Not Feasible
41 - 60	Enough
61 - 80	Feasible
81 - 100	Very Feasible

The maintainability aspect was tested by calculating the value of the Maintainability Index (MI) which consists of three components, namely the Line of Code (LoC), Cyclomatic Complexity (CC), and Duplication Source Code. The LoC measurement phase was done by calculating the source code manually, which is the calculation of the size of the software made. The size of the software created is a factor for diagnostic assessment and analysis requirements in software using the Line of Code (LoC) metric (Koyya, Lee, & Yang, 2013, p. 3). The process of calculating the Duplication Source Code was done using Gendarme 2.11 software. Calculation of the value of Maintainability Index (MI) was processed using software that is used to compile C# language code scripts in the code generation process that has an analyzer feature. The value of MI that has been obtained was translated into the care classification (Microsoft Developer, 2007), as shown in Table 3.

Table 3. MI Value Classification

Maintainability Index (MI)	Classification	Symbol
0 - 9	Difficult Maintain	
10 - 19	Medium Maintain	
20 - 100	Easy Maintain	

Testing the portability aspect consists of the installation, uninstalling, operating on various kernel versions of the Android OS, and

sharing the hardware configuration. The instrument used is a checklist to record test results. The usability aspects were tested using instruments in the form of questionnaires given to the users. Testing in measuring the usability level of software is a beta testing aspect (Chemuturi, 2011, p. 183). The usability test questionnaire used was the USE Questionnaire (Lund, 2001, p. 4). The USE Questionnaire is highly recommended because it contains various aspects including Usefulness, Satisfaction, Ease of Use, and Ease of Learning and has a Likert scale for its level (Tullis & Albert, 2013, p. 142). The assessment technique for usability aspects uses a Likert scale, because it is related to the activities of measuring attitudes, opinions, and perceptions of a person or group about social events or symptoms, by giving five choices of scale responses (Riduwan, 2013, p. 13) (see Table 4).

Table 4. Likert Scale

Alternative Answers	Score
(STS) Very Disagree	1
(TS) Disagree	2
(N) Neutral	3
(S) Agree	4
(SS) Very Agree	5

The reliability test of the USE Questionnaire questionnaire was conducted using the Cronbach Alpha analysis method. The criteria for a research instrument are said to be reliable if the reliability coefficient index number is  $(r_{11}) > 0.6$  (Arikunto, 2013, p. 319).

Testing the media expert stage was done to find out the quality of the media from the perspective of media experts. The characteristics of media that are good in mobile learning are having characteristics that are in accordance with the device in question (mobile tools), namely mobile devices that have the characteristics of small size and small screen resolution, of course, requires software that is adjustable (Pachler, Bachmair, & Cook, 2010, p. 68). The characteristics of such mobile devices lead to indicators and instrument grids of software media specifications that must be oriented towards adjusting the user interface design and visual appearance, namely in the form of images, symbols, colors, buttons, animations, letters, which correspond to the characteristics of mobile devices which, mostly, are

small. After the grid of instruments for evaluating the media expert has been compiled, it was subsequently confirmed in the form of a statement. Testing the aspects of material experts is based on learning material and the characteristics of digital engineering learning devices.

## RESULTS AND DISCUSSION

### Requirement Modeling

The design activity of digital engineering learning software "DigiChip" begins with the collection of information and data about product detail specifications that represent the user's needs. The process carried out includes: (1) observing the teaching and learning process when using digital engineering hardware devices to determine the level of weakness in the use of hardware media, the level of danger, and risk during the teaching and learning process, (2) conducting interviews with students, laboratory/workshop managers and teachers, about difficulties and weaknesses the use of teaching and learning methods that have been carried out so far, such as the problem of students not being able to study independently at home because of limited equipment which is only available at school, the problem of expensive digital engineering learning devices if students must have it to study independently at home, digital learning activities that must take turns or groups because of the limited digital engineering equipment, (3) discussions about effective and efficient media specifications for the needs of students and teachers in the learning process of digital logic gate techniques.

The results of studying various problems from the information that has been obtained show that, (1) scientifically, the working mechanism of an IC (Integrated Circuit) cannot be understood only by looking at the external physical form, as well as the IC logic gate in digital techniques that have many system variants work. The solution to overcome this problem is the concept of a learning media product which can simulate the working mechanism of the real hardware. The concept of digital engineering learning media products is in the form of software that is packaged in devices that are easily accessible/owned by students, flexible, and portable. (2) The software which is made can be operated on smartphone

devices, not on laptops or PCs because almost all students have a smartphone, even more than one unit, but not all students have a laptop or PC. (3) The software made accommodates the basic needs of digital engineering subjects, such as material about six logic gates base (NOT, AND, OR, NAND, NOR, XOR), providing truth tables as auxiliary material facilitates the logic gate learning process. (4) Students can operate various working mechanisms from six basic logic gate systems, for students to understand the characteristics of the system.

**Architectural Design**

The design process consists of; (1) User Experience (UX) Design, and (2) User Interface (UI) Design. The UX design process for this "DigiChip" product uses the UML language as a system design model. The system mechanism in this research product is described in the three main diagrams of the modeling language, namely (a) use case diagrams, (b) activity diagrams, and (c) sequence diagrams. The scheme in the use case diagram contains the working mechanism of the "Digi-Chip" product system. The use case diagram scheme displays the user's mechanism to operate various features designed in the "Digi-Chip" software for mobile devices. Figure 2 shows the design of a use case diagram from the "DigiChip" software.

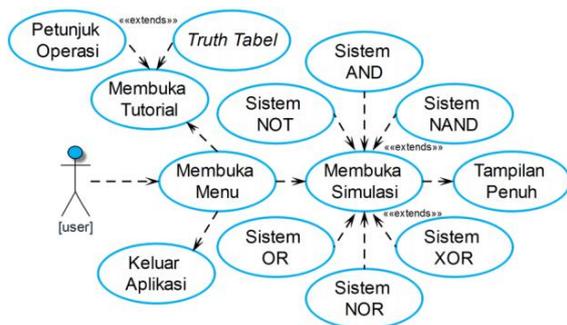


Figure 2. Use Case Diagram

The next process in designing learning media software Digital Techniques "DigiChip" is making activity diagrams, as presented in Figure 3. An activity diagram explains how a software works on instructions or stimuli provided by the user so that there is an interaction between the user with the machine (mobile device) after the expected plan (see Figure 4).

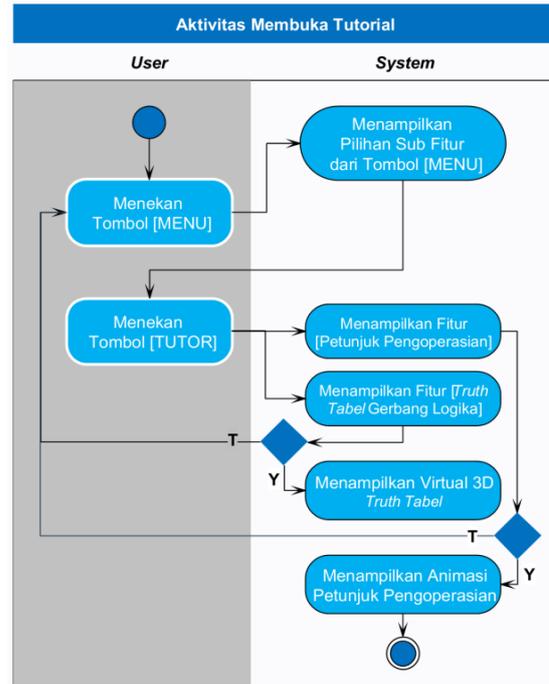


Figure 3. Activity Diagram of Tutorial

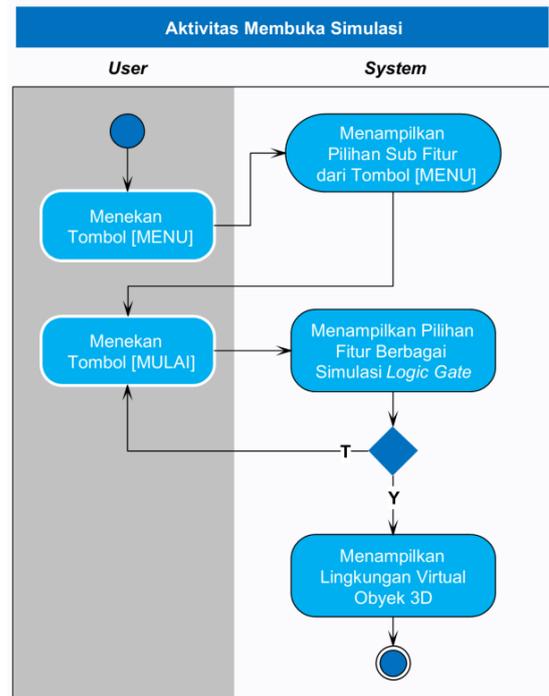


Figure 4. Activity Diagram of Simulation

The next step is designing sequence diagrams (Figure 5), which function as modeling languages that describe user activities in interacting with machines/systems. User interaction activities with the system are described in a timeline.

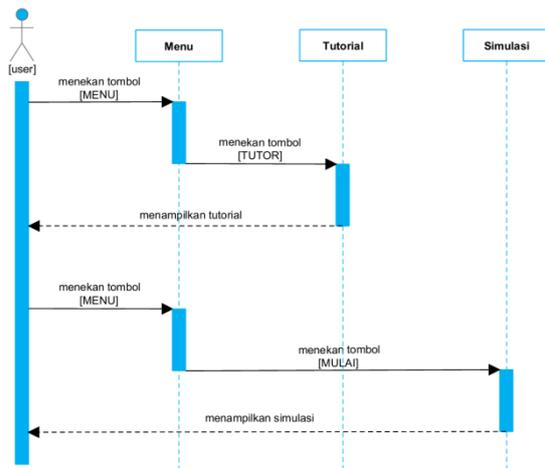


Figure 5. Sequence Diagram

The next stage of the process in architectural design is the design process of the User Interface (UI) to adjust the user's comfort when interacting with the mobile device through the touchscreen. Ergonomics of operational convenience refers to the placement of various positions of UI components on the screen so that it is convenient to operate with one or two hand grips. The placement of various UI component positions in the "DigiChip" software is designed based on other consideration factors, namely the consideration that the majority of users have a culture of right-handed users, not left-handed. The UI in the "DigiChip" software consists of various pages/screens, including (a) splash screen, (b) home screen, (c) simulation screen, and (d) tutorial screen (see Figure 6 & Figure 7).

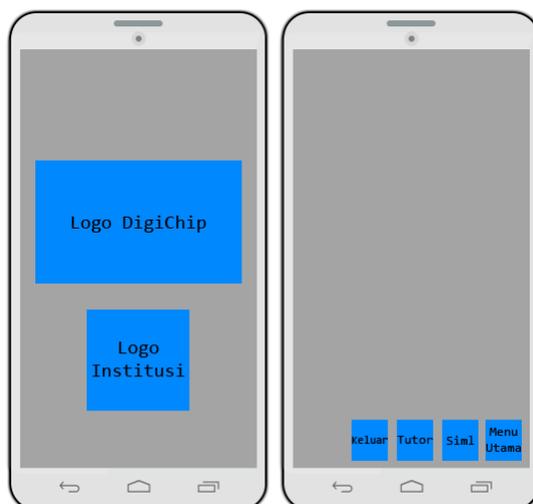


Figure 6. UI Screen – Splash & Home

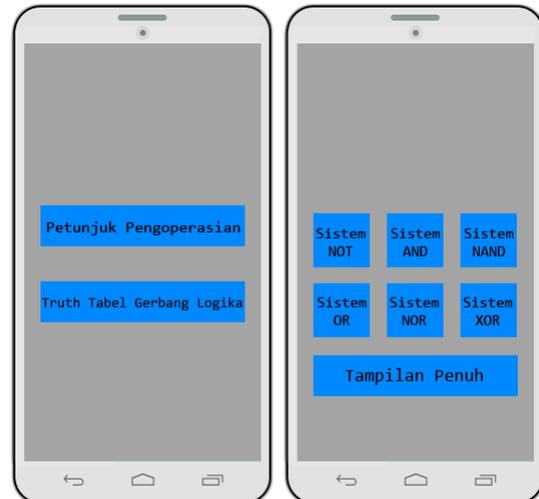


Figure 7. UI Screen – Simulation

### Component Design

The next process in designing "Digi-Chip" software is component design, which is the activity of designing various 2D and 3D components needed to realize "DigiChip" Digital Engineering learning media software (Figure 8). Component design activities involve around four software builders in the form of specialties for 2D object design, 2D object coloring, 2D skin making, 3D shape designers, 3D object coloring, and 3D textures and materials.



Figure 8. Splash-Home Screen

Various components of 3D objects in "DigiChip" software such as ICs, IC sockets, LEDs, LED ports, switches, are designed to have dimensions and scale ratios exactly the same as the actual object components, with



the script (coding). The process for knowing the performance mechanism of the source-code through white-box testing by creating a workflow base path, and first making a flowgraph. Before making the flowgraph, a flowchart is first made (see Figure 11 and Figure 12). A flowchart is designed as an auxiliary graph of the flowgraph to make it easier for the reader to understand because the flowgraph only contains numeric codes on a base path which the reader must understand for longer (Table 5).

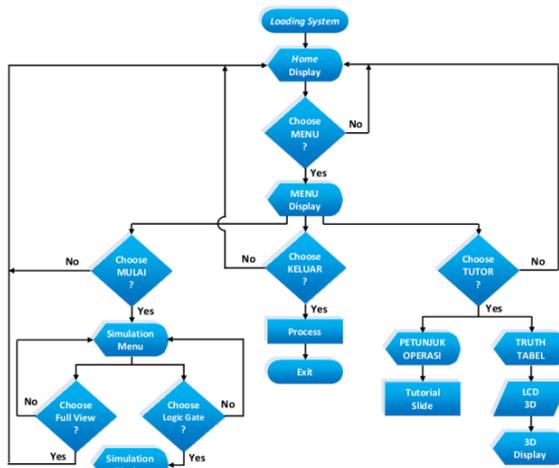


Figure 11. Flowchart

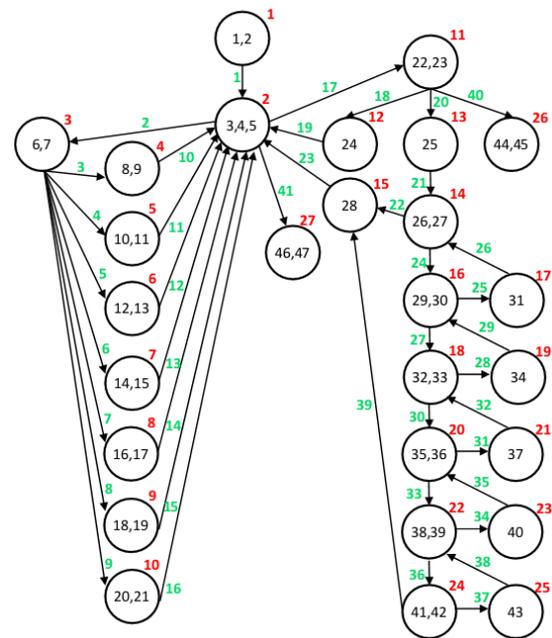


Figure 12. Flowgraph

Table 5. Flowgraph Description

No.	Description	No.	Description
1.	Start	24.	Exit TUTOR
2.	Splash screen loading system	25.	Touch Operating Instructions
3.	Home page	26.	Appearance setting
4.	Touch MENU	27.	Next the appearance setting page
5.	START, TUTOR, EXIT page	28.	Exit the Operating Instructions
6.	Touch START	29.	Displaying Scematic
7.	Simulation options display page	30.	Next Displaying Scematic
8.	Touch system NOT	31.	Prev Displaying Scematic
9.	Logic gate simulation NOT	32.	Plug the connector wire
10.	Touch system AND	33.	Next the Plug connector wire
11.	Logic gate simulation AND	34.	Prev the Plug connector wire
12.	Touch system NAND	35.	Unplug the connector wire
13.	Logic gate simulation NAND	36.	Next the Unplug the connector wire
14.	Touch system OR	37.	Prev the Unplug the connector wire
15.	Logic gate simulation OR	38.	Input button
16.	Touch system NOR	39.	Next the Input button
17.	Logic gate simulation NOR	40.	Prev the Input button
18.	Touch system XOR	41.	LED Output
19.	Logic gate simulation XOR	42.	Next LED Output
20.	Touch fullscreen	43.	Prev LED Output
21.	Exit simulation options	44.	Touch the Truth Table
22.	Touch TUTOR	45.	Truth Table display page
23.	Tutorial options display page	46.	Touch EXIT

The next process in unit testing is first to calculate Cyclomatic Complexity (CC). Calculations are denoted by  $V(G) = e - n + 2$ . Notation "e" is the number of edges (system speed), and "n" is the number of nodes (system features). The value of CC is used for the next process, which is determining the independent path. The "DigiChip" software that has been created has 41 edges and 27 nodes. Then CC or  $V(G)$  is as shown in Equation 2:

$$\begin{aligned} e &= 41 ; n = 27 \\ V(G) &= e - n + 2 \\ V(G) &= 41 - 27 + 2 = 16 \end{aligned} \quad (2)$$

The next process is to determine the independent path, which is a representation of the speed of the system when governed by the user from the beginning to the end — the total number of lines as calculated in the CC calculation, which is 16 (Table 6).

Unit testing consists of 47 execution features available in the "DigiChip" learning media software. Feature items that operate well are recorded in the feature test list with a check list. Data that has been obtained from the unit testing stages are analyzed based on the feasibility percentage formula (Equation 3).

$$\text{Feasibility percentage (\%)} = \frac{\text{Total score obtained}}{\text{Total highest score}} \times 100 \quad (3)$$

The calculation results obtained are then converted to the feasibility category. 100% unit testing is included in "Very Worthy" category.

### **Integration Testing**

The next test that must be done to improve the results of the software design that has

been made is integration testing, which is to run the system/program algorithm by referring to the list of test cases made. Integration testing is done using the black-box method. The integration testing process is carried out when all elements in the previous stage, such as architectural design, user interface, and code script, have been integrated. The integration testing stage is conducted to find out whether the "DigiChip" algorithm and systematics of software made are in accordance with the planning stage without any errors. The results of integration testing are used as a reference in determining the functional suitability aspects of the ISO/IEC 25010 software quality standard, which focuses on the suitability of a single function to be able to perform certain tasks.

Integration testing is in the form of testing the operation of various features through a list of 16 test cases using the black-box method. Calculation analysis obtained is presented in Equation 4.

$$\begin{aligned} \text{Operational test case (\%)} &= \frac{\text{Total score obtained}}{\text{Total highest score}} \times 100\% \\ &= \frac{16}{16} \times 100\% \\ &= 100\% \end{aligned} \quad (4)$$

The calculation results obtained are then converted to the feasibility category table. The percentage of 100% integration testing is in the "Very Worthy" category. Other meanings can be interpreted that 100% nominal explains that all the features of the "DigiChip" learning software that have been made run well.

Table 6. Flowgraph Description

No.	Independent Path
1.	1,2→3,4,5→6,7→8,9
2.	1,2→3,4,5→6,7→10,11
3.	1,2→3,4,5→6,7→12,13
4.	1,2→3,4,5→6,7→14,15
5.	1,2→3,4,5→6,7→16,17
6.	1,2→3,4,5→6,7→18,19
7.	1,2→3,4,5→6,7→20,21
8.	1,2→3,4,5→22,23→24
9.	1,2→3,4,5→22,23→25→26,27→28
10.	1,2→3,4,5→22,23→25→26,27→29,30→31
11.	1,2→3,4,5→22,23→25→26,27→32,33→34
12.	1,2→3,4,5→22,23→25→26,27→35,36→37
13.	1,2→3,4,5→22,23→25→26,27→38,39→40
14.	1,2→3,4,5→22,23→25→26,27→41,42→43
15.	1,2→3,4,5→22,23→44,45
16.	1,2→3,4,5→46,47

### System Testing

The next stage is system testing, which is the stage of software testing that are made when between parts have been incorporated into a single unified system. System testing is testing the overall performance of software products created. This process is carried out by the engineer/programmer concerned. The system testing process uses black-box technique, by calculating (1) the Line of Code (LoC) source code, (2) Duplication Source-code, and (3) The insulation test. Calculation of Line of Code (LoC) is an accumulation process of the amount of source code used in making "Digi-Chip" learning media software. Calculation of source code is done manually by accessing one by one script file embedded in the object, in the graphics engine software used (see Figure 13). The results of calculating the total number of source code used are 10,541 lines of source code or LoC (Line of Code). All source code in script files is written in C# (Csharp).

The next testing process is the calculation of the number of duplication source code in all script files containing 10,541 lines of source code. The process of calculating and analyzing duplication source code to be more precise and without errors was done using a software analyzer system which is in one type and brand with the system used in the process of designing source code; it is to minimize the occurrence of analyzed errors and weak accuracy due to differences in system formats. The results of the duplication source code analysis show there are two lines of source code from 10,541 lines of source code created. The 2-line duplication source code is in a script file called `RejTouchEv.cs`, which is a C# language file extension.

System testing was done to determine the maintainability and portability aspects of the ISO 25010 standard. The maintainability aspect is expressed by calculating the Main-

tainability Index (MI) value on all source code files used in making "DigiChip" learning media. MI values were calculated using software used to compile C# language code scripts in the process of creating "DigiChip" learning media, which has an analyzer feature. The results of the MI values calculation are shown in Table 7.

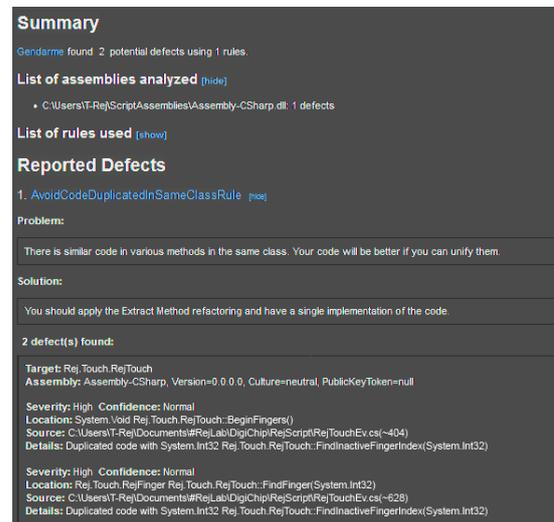


Figure 13. Duplication Source Code Analysis

Table 7. Maintainability Index (MI) Total

Solution	'DigiChip MI' (1 project)	Maintainability Index
Assembly-Csharp (Debug)		84

The known MI value is 84 with a green symbol; if translated referring to Table 3, it means that the entire code script in the "Digi-Chip" software has "easy maintenance" category. In theory, all future code scripts will be easy to develop and modify to the desired innovation stage.

The next testing process is to test the installation of various types of mobile device hardware configurations. The installation test process is basically more about the operational analysis of software that has been made in various kernel variants or the Android operating system (OS) version (see Table 8).

Table 8. Portability Testing

No.	OS Version	Type	Install	Running	Result Score	Max. Score
1	4.2 Jelly Bean	Samsung Galaxy G-Neo	1	1	2	2
2	4.4 Kit kat	Lenovo A536	1	1	2	2
3	5.0 Lolipop	Asus ZenFone 2 Laser	1	1	2	2
4	6.0 Marshmallow	Asus ZenFone 3	1	1	2	2
5	7.0 Nougat	Xiaomi Redmi Note 4 Pro	1	1	2	2
6	8.0 Oreo	Vivo Y71	1	1	2	2
Total Score					12	12

Screen ratio parameters are not a priority because the "DigiChip" software created was designed with the ability to auto-extend at various screen ratios. The characteristics of the "DigiChip" software is a realtime 3D simulation software, which requires the specifications of mobile devices that have adequate graphics capabilities. The software products have the character of driving graphics hardware performance continuously as long as the application is "on", and it will stop when it is "off". Such software characters require different hardware performance compared to other software products that have static performance characteristics. Thus, the testing at the installation test stage focuses on the graphic capabilities of mobile devices to run the "DigiChip" Digital Engineering learning media software.

The method used in testing the portability aspect is the running test technique owned by an emulator device, such as an AVD (Android Virtual Device). The type of configuration of the smartphone device and the kernel and Android OS variants was tested based on user-sourced data. The results of the test on various configurations of mobile devices have not found any errors in the install process or during the operation of "DigiChip" software continuously. Testing on a variety of mobile devices that have CPU (central processing unit) and GPU (graphics processing unit) hardware configurations is the least even if there is no system bottleneck or errors during the running.

Portability aspects testing was done using the installation process on various smartphone device configurations. The installation test is carried out on six Android OS variants. Based on the portability test results shown in Table 8, it can be calculated with the formula presented in Equation 5.

$$\begin{aligned} \text{Portability Testing (\%)} &= \frac{\text{Total score obtained}}{\text{Total highest score}} \times 100\% \\ &= \frac{12}{12} \times 100\% \\ &= 100\% \end{aligned} \quad (5)$$

The calculation results that have been obtained were then converted to the feasibility category. The 100% portability testing was included in the "Very Worthy" category.

### **Acceptance Testing**

The acceptance testing process involves the assessment of experts, in this case, material

experts and media experts. The next process is the assessment of the respondents as the end-user of the DigiChip Digital Engineering learning media software product. Acceptance testing instruments are prioritized by expert judgment. The assessment carried out by material experts is an aspect to determine the quality of the "DigiChip" software product from the parameters of conformity to the content of the material with teaching materials designed from the school. The assessment of media experts is an aspect of the assessment of learning media from the quality of software products that have been made, more specifically about the point of view of a software to be worthy of being called a learning tool or media in the learning process. The results of testing on media experts obtained a value of 63. Then calculated as the percentage of eligibility (Equation 6).

$$\begin{aligned} \text{Eligibility Percentage (\%)} &= \frac{\text{Total score obtained}}{\text{Total highest score}} \times 100\% \\ &= \frac{63}{70} \times 100\% \\ &= 90\% \end{aligned} \quad (6)$$

The calculation results of the eligibility percentage from the media expert test were then translated based on the feasibility table, and the results are included in the category of "Very Eligible". Furthermore, the test results on material experts obtain a value of 19. Thus, it is calculated as the percentage of eligibility as follows (Equation 7):

$$\begin{aligned} \text{Eligibility Percentage (\%)} &= \frac{\text{Total score obtained}}{\text{Total highest score}} \times 100\% \\ &= \frac{19}{19} \times 100\% \\ &= 100\% \end{aligned} \quad (7)$$

The calculation results of the eligibility percentage from the material expert test, if interpreted based on the feasibility table, were then categorized in "Very Worthy". Meanwhile, the calculation of the feasibility percentage of usability testing aspects is as follows (Equation 8).

$$\begin{aligned} \text{Feasibility Percentage (\%)} &= \frac{\text{Total score obtained}}{\text{Total highest score}} \times 100\% \\ &= \frac{7119}{8260} \times 100\% \\ &= 86.18\% \end{aligned} \quad (8)$$

The calculation results of the feasibility test percentage on 59 users are then interpreted according to the feasibility table, which is cate-

gorized into "Very Worthy". The reliability parameter analysis on the aspects of usability testing was performed using Cronbach Alpha ( $\alpha$ ) alpha counting software. The results obtained are presented in Table 9.

Table 9. Alpha Cronbach Calculation

Case Processing Summary			
		N	%
Cases	Valid	59	100.0
	Excluded <sup>a</sup>	0	.0
	Total	59	100.0

<sup>a</sup>. Listwise deletion based on all variables in the procedure

Reliability Statistics	
Cronbach's Alpha	N of Items
.841	28

Cronbach's Alpha 0.841 is then interpreted through the description table (George & Mallery, 2016, p. 240) shown in Table 10. The results of alpha calculations that have been obtained (0.841) can be interpreted as "good".

Table 10. Alpha Value Description IBM SPSS

$\alpha > .9$ – excellent
$\alpha > .8$ – good
$\alpha > .7$ – acceptable
$\alpha > .6$ – questionable
$\alpha > .5$ – poor
$\alpha < .5$ – unacceptable

## CONCLUSION

The results of a series of research, design, and testing of "DigiChip" learning media products are as follows: (1) "DigiChip" software as a learning tool for Digital Engineering Subjects in science Mechatronics is developed through the SDLC type V-model. V-model procedures and stages include Requirement Modeling, Architectural Design, Component Design, Code Generation, Unit Testing, Integration Testing, System Testing, and Acceptance Testing. The next stage of testing in the software that has been made is to use the SQA procedure using the ISO/IEC 25010 software testing standard. (2) "DigiChip" software testing parameters using ISO/IEC 25010 consists of Functional Suitability, Maintainability, Portability, and Usability aspects. The results of a series of testing processes conclude that in the functional suitability aspect shows the percentage of 100% feasibility of all features oper-

ating properly, with the classification of the feasibility table "Very Worthy". The test results on maintainability aspects show the value of the Maintainability Index (MI) of 84, which means in the category "Easy Care". The portability test results show that the percentage of 100% "DigiChip" learning software can operate on various variants of the Android OS, including the "Very Worthy" category. The usability test results show 86.18% in the "Very Worthy" category, with a reliability value of 0.841 included in the "Good" category. The stages of testing aspects of media experts show that 90% belong to the "Very Worthy" category. Testing aspects of material experts show that 100% are included in the "Very Worthy" category.

## REFERENCES

- Arikunto, S. (2013). *Prosedur penelitian: Suatu pendekatan praktik*. Jakarta: Rineka Cipta.
- Chemuturi, M. (2011). *Mastering software quality assurance: Best practices, tools and techniques for software developers*. Plantation, FL: J. Ross.
- George, D., & Mallery, P. (2016). *IBM SPSS statistics 23 step by step: A simple guide and reference* (14th ed.). Oxon: Routledge.
- Heitlager, I., Kuipers, T., & Visser, J. (2007). A practical model for measuring maintainability. In *6th International Conference on the Quality of Information and Communications Technology*. Lisbon, Portugal: IEEE Explore. <https://doi.org/10.1109/QUAT IC.2007.8>
- Koyya, P., Lee, Y., & Yang, J. (2013). Feedback for programming assignments using software-metrics and reference code. *International Scholarly Research Notices (ISRN) Software Engineering, 2013*, 1–8. <https://doi.org/10.1155/2013/805963>
- Laplante, P. A. (2007). *What every engineer should know about software engineering* (1st ed.). Boca Raton, FL: CRC Press.
- Lund, A. M. (2001). Measuring usability with the USE Questionnaire. Retrieved May

- 12, 2016, from <https://www.researchgate.net/publication/230786746/>
- McCabe, T. J. (1976). A complexity measure. *IEEE Transactions on Software Engineering*, SE-2(4), 308–320. <https://doi.org/10.1109/TSE.1976.233837>
- Microsoft Developer. (2007). Maintainability index range and meaning – Code analysis team blog. Retrieved May 9, 2017, from <https://blogs.msdn.microsoft.com/codeanalysis/2007/11/20/maintainability-index-range-and-meaning/>
- Najadat, H., Alsmadi, I., & Shboul, Y. (2012). Predicting software projects cost estimation based on mining historical data. *International Scholarly Research Notices (ISRN) Software Engineering, 2012*, 1–8. <https://doi.org/10.5402/2012/823437>
- Pachler, N., Bachmair, B., & Cook, J. (2010). *Mobile learning: Structures, agency, practices*. (G. R. Kress, Ed.). London: Springer.
- Pressman, R. S. (2010). *Software engineering: A practitioner's approach* (7th ed.). New York, NY: McGraw-Hill.
- Riduwan. (2013). *Skala pengukuran variabel: Variabel penelitian*. Bandung: Alfabeta.
- Tamrin, A. G., Slamet, S., & Soenarto, S. (2018). The link and match of the demand and supply for productive vocational school teachers with regard to spectrum of vocational skills in the perspective of education decentralization. *Jurnal Pendidikan Vokasi*, 8(1), 40–52. <https://doi.org/10.21831/jpv.v8i1.15135>
- Tullis, T., & Albert, B. (2013). *Measuring the user experience: Collecting, analyzing, and presenting usability metrics* (2nd ed.). Waltham, MA: Elsevier.
- Watkins, J., & Mills, S. (2011). *Testing IT: An off-the-shelf software testing process* (2nd ed.). Cambridge: Cambridge University Press. <https://doi.org/10.1017/CBO9780511997310>
- Williams, L. (2006). Testing overview and black-box testing techniques. Retrieved May 12, 2016, from <http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf>.