# Instal : Jurnal Komputer

**E-ISSN: 2808-683X**
**Edisi : Volume 16 Nomor 03 | 2024**

Available online at https://journalinstal.cattleyadf.org/index.php/Instal/index

## Anxiety in Programming Course of University Students: Does It Affect Students' Performance?

Tansa Trisna Astono Putri[1], Reni Rahmadani[2], Hanapi Hasan[3]
[1,2] Information Technology and Computer Education Study Program of Universitas Negeri Medan, Indonesia
[3]Mechanical Engineering Education Department of Universitas Negeri Medan, Indonesia

***Keywords:***

Undergraduate students, programming, anxiety, performance, Computer and Electrical Engineering

**ABSTRACT**

In the context of programming, anxiety typically refers to the feeling of unease or worry that can arise when working on software development tasks or projects. It is a psychological and emotional response to the challenges and uncertainties that programmers often face. In this study, undergraduate students of Computer and Electricical Engineering often work under strict deadlines and face pressure to deliver high-quality code. In this study, undergraduate students often work under strict deadlines and face pressure to deliver high-quality code. This can lead to anxiety as they strive to meet expectations and complete tasks on time. Programming involves solving complex problems, and sometimes the solutions are not immediately apparent. This can create anxiety as programmers grapple with uncertainty and the fear of making mistakes for undergraduate students. The purpose of this quantitative study was to look into anxiety as the root cause of students' poor programming performance. The results of this study will aid in providing the most effective approach for instructing and learning programming courses in order to generate competent programmers for use in business settings and higher education. Data for this quantitative study were gathered via the questionnaire. The study found that the degree of unfavourable beliefs about control in computing scenarios and the low level of computing self-efficacy found in this study indicate that computer programming course designers must come up with ways to boost students' confidence and motivation.



***Corresponding Author:***
**Tansa Trisna Astono Putri**
Universitas Negeri Medan
Email: tansatrisna@unimed.ac.id

## INTRODUCTION

These days, programming is crucial since information systems are used to support people in many facets of their lives. Our lives have been significantly impacted by programming and its

Page : 445-452
Doi. https://doi.org/10.54209/jurnalinstall.v16i03.284

technologies, which include web applications, games, social media, online communication, and cloud storage. In order to create a system, programming must take advantage of technological advancements like artificial intelligence, machine learning, virtual and augmented reality, mobile programming, the Internet of things, and more. By employing technology, we are able to create any kind of technology for any kind of aspect of life. Education is one area where technology is being used.

Students' need for programming skills is growing in the current environment. In a variety of industries, including software development, banking, healthcare, and even entertainment, the ability to programme is highly valued. Students who study programming may find a wide range of employment opportunities. Students need to think critically and creatively in order to solve problems. This ability is helpful in many other facets of life besides computer science. The ability to programme will become more and more crucial as technology advances and permeates every aspect of our lives. Students who study programming now can prepare themselves for the workforce in the future. By breaking down complex problems into smaller, more manageable parts, computational thinking—which is facilitated by learning to code—can be developed in students. This skill can be applied to a wide range of academic and real-world contexts. By enabling students to create their own websites, games, and programmes, programming may give them a creative outlet. This can help them become more creative and proficient at solving problems.

According to a study [1], as many as 65% of college students fail to pass the course and only a small percentage receive full marks. According to [2], over 30% of undergraduate computer science students worldwide dropped out or performed poorly in their programming courses. [3] linked the diploma's cancellation to the introductory program's non-release. Institutions, educators, and students must be very concerned about these alarming dropout and failure rates. The desire of students to eschew courses, which are necessary for all computer majors at universities, will cause problems because it will delay graduation owing to failure. Teachers put in a lot of work and time in lectures and labs with college students to help them understand programming better, but given the high costs associated with failure and attrition, these efforts may be viewed as pointless. Some Indonesian university students experience similar difficulties when taking programming courses. The difficulties that students encounter when it comes to understanding basic programming since they are no longer familiar with a particular programming language [4]. In addition, they need to become proficient in three interconnected areas: programming language syntax, design, and programming structure. The same challenges are encountered by Indonesian public university students when taking programming courses. Students pursuing a major in electronics engineering must pass the programming course. Beginning college students face a challenging task when learning programming. It's not just about learning the syntax of programming languages. It also involves the ability of college students to develop a set of guidelines that could potentially solve a particular problem [5]. Some fundamental ideas that students may encounter in class include variables, arrays, and iteration. In an attempt to pass an assignment, students often engage in a few horrible programming behaviours.

There is a study programme in computer education and information technology at a public university in North Sumatra. The study programme for Information Technology and Computer Education is a component of the Electronics Engineering major. The bachelor's degree offered by that programme is in computer science and informatics. Students study basic programming from the first semester to the seventh semester, when they are expected to develop an entire information system. Students must also learn about databases, programming structures, and system architecture, including language programming and server, in order to construct a comprehensive information system. The concept of the system, as well as the requirements for the application—web or desktop—must be decided upon by the students. Thus, students' fundamental programming concepts must be solid.

Page : 445-452
Doi. https://doi.org/10.54209/jurnalinstall.v16i03.284

According to the researcher's experience teaching programming and interviews with lecturers, students frequently err when creating the right code for an issue that is currently facing them. They are writing a line that they do not understand. They simply write the code without even understanding the concepts involved, such as variables, which may vary depending on the problem at hand. There have been cases where university students have studied large portions of software code by rote, understanding very little or nothing at all. Additionally, some college students composed lengthy software code devoid of syntax and logical testing, leading to a wide range of errors and deterring the students from programming entirely [6]. Additionally, they are ignorant of looping, iteration, and array usage. As a result, they are unable to appropriately handle an error exception that the programming editor throws.

In the context of programming, anxiety typically refers to the feeling of unease or worry that can arise when working on software development tasks or projects. It is a psychological and emotional response to the challenges and uncertainties that programmers often face. In this study, undergraduate students often work under strict deadlines and face pressure to deliver high-quality code. This can lead to anxiety as they strive to meet expectations and complete tasks on time. Programming involves solving complex problems, and sometimes the solutions are not immediately apparent. This can create anxiety as programmers grapple with uncertainty and the fear of making mistakes for undergraduate students.

It has been shown that computer anxiety affects computer skills [7]. Research by [8] also showed a negative correlation between maths anxiety and academic performance. According to other studies, anxiety related to mathematics has a negative effect on one's attitude towards computers, which is likely to have a big effect on how well one programmes computers [7]. In any programming course, students who are more at ease with programming tasks perform better than those who are a little nervous about it. Therefore, it stands to reason that the least amount of worry—whether it be related to computers, programming, or mathematics—should be maintained in order to achieve successful programming outcomes.

A. Anxiety in Programming

It might be challenging for first-time programming students to build strong psychological representations of how a programme functions. The complexity of developing their coding abilities beyond what they have previously taught is another challenge. Students' distaste for computer programming has increased along with the changes in the content of introductory computer science courses. The ordering of basic operations made up programming language when computers were first conceived and operated manually. After procedural abstraction and object orientation were introduced, computer thought grew increasingly sophisticated. Because of this, programmes allow a sequence of these processes to be handled as a new collection rather than altering data in a single, straightforward step, which makes some students uneasy or even scared of programming. Anxiety negatively impacts academic achievement, which in turn impacts retention rates.

Particularly elevated, unregulated levels of anxiety have a negative impact on children's academic performance, resulting in low student achievement and high dropout rates [9]. According to [10], computing anxiety is a mental disorder that occurs while a learner encounters or anticipates experiencing a loss of self-worth in a computing environment. It is also a powerful predictor of a student's proficiency with computers [11]. Additionally, this kind of anxiety significantly impairs students' capacity to use computers efficiently, and although experience lessens fear, different levels of computer anxiety endure [11]. Speier found a connection between low skill performance and high anxiety levels at the beginning of computer learning [12].

An inaccurate or dysfunctional perception of a situation is what keeps anxiety levels high. As a result of an inaccurate evaluation of their aptitude for learning computer programming, pupils suffer from programming anxiety. This is due to the fact that the students' conceptual frameworks are unable to be utilised to dissect the programming issues and come up with an

Page : 445-452
Doi. https://doi.org/10.54209/jurnalinstall.v16i03.284

alternative, and the activation process cannot take place [13]. Most first-year college students are not very good at computer programming, and their cognitive models may not be as developed as they should be. A few studies on the factors influencing a student's capacity to acquire knowledge in programming have been conducted [14].

But a number of those talents may not have been fully developed prior to beginning their undergraduate degree programme, which would make the student nervous when they have to learn programming. In addition to feeling inadequate, a student who receives negative feedback from the computers, for example, because their programmes won't assemble, becomes anxious. According to McInereny's description of computer anxiety, circumstance anxiety is defined for the purposes of this study as programming anxiety, which is defined as a psychological condition produced if a student experiences or anticipates losing consciousness when faced with a computer programming circumstance [15].

## RESEARCH METHOD

**Table 1.** Demographic

| Demographic | | Computer and Informatics Education | Electrical Engineering |
|---|---|---|---|
| Gender | | | |
| | Male | 47 | 64 |
| | Female | 38 | 7 |
| Age | | | |
| | 17 | 8 | 10 |
| | 18 | 56 | 38 |
| | 19 | 18 | 16 |
| | 20 | 3 | 7 |
| Total | | 85 | 71 |

The research population for this study was selected using a non-probability sampling technique that is representative [16]. The participants in this study are Indonesian university students, for whom programming is a prerequisite course. For the purposes of this study, a sample of the Indonesian population will be selected from Medan. The university, Universitas Negeri Medan, is then selected through purposive sampling since it possesses the quality required for the sample for this study. In this study, the university will be selected through purposeful sampling, and the undergraduate study correspondents will be selected through random sampling. This study will only examine college students. Every first year student from Electrical Engineering and Informatics and Computer Education major will be placed in a class with a minimum of thirty other students of various genders and ethnicities. Consequently, the total sample size would be approximately 156 students. From Table 1, we could see that the demographic of this study is not limited to only one gender. The number of students of each gender also not arranged by the researcher. Most of the students in Electrical Engineering are male. The mean of students' age is 18.3 year. In this study, a random sampling technique will be used to select the undergraduate participants.

Not only are the students in the programming course involved, but the lecturers' assistance in this study is widely available and controllable, so representative sampling will be used. The Anxiety in Computer Programming Questionnaire would be utilised to evaluate anxiety. The computer programming anxiety questionnaire (CPAQ) examined several topics, including student characteristics, goal orientation, experience learning computer skills, perceived control, computer self-concept, and anxiety levels in computing-related situations [15].

Page : 445-452
Doi. https://doi.org/10.54209/jurnalinstall.v16i03.284

**RESULTS AND DISCUSSION**

92% of all first-year students enrolled took part in the initial Pre-CPAQ during the first evaluation cycle, which took place in Academic Year 2023–2024. The group's mean age was 18.3 years. The following sections present data indicating the level of anxiety in each of the four CPAQ classifications (12 variables and 72 phrases) for the group of participants during the academic year 2023-2024.

The researcher organised the questions in the questionnaire to allow for a range of answers. Students are asked to rate how strongly they agree, disagree, are undecided, agree, or disagree with a series of statements using a Likert scale. The level of anxiety in computer circumstances aspect used a four-point response scale because it proved preferred for assessing anxiety frequency in four categories (resulting in no room for confusion about rare anxiety).

A. Gaining Initial Computing Skills

The CPAQ's first classification, "Gaining Initial Computer Skills," included 22 phrases covering four different variables that related to computer-related impression that could make people nervous or uneasy. The four components were: 1) proficiency with computers; 2) equipment handling; 3) feedback on computing abilities; and 4) education on fundamental computer operations. The area where the most anxiety was displayed was when it came to demonstrating computer competency. Regarding learning computer functions, the statement that indicated dealing with computer malfunctions caused the most anxiety (27.5%) was concerning for computing skills, obviously.

All four Post-CPAQ factors show a decrease in anxiety related to Gaining Initial Computing Skills, which is encouraging. The median anxiety level in handling computer equipment also declines, with a decrease from 2.59 pre to 2.10 post (IQR= 1.07) in competency with computers. Similar findings are obtained when learning about basic computer function and getting feedback on one's computing skills, as Table 2 illustrates.

B. Sense of Control

Two variables were found to be associated with both a positive and negative sense of control when utilising a computer. The elements are a reflection of how people perceive themselves to be powerless over their own lives, as seen in their thoughts. A sizable fraction of the students polled expressed little or no sense of positive control over developing their computer skills. The second factor (fear/negative) indicates the extent or absence of perceived personal control over computing, expressed as fear or negative self-talk. Phrases like "What if I hit the wrong key?", “People will notice if I make a mistake” and "I'm afraid I'll wreck the program/hard drive," which are typical of phobic reactions to computers, all show very slight anxiety.

The results of the postcomputer anxiety study show a significant decrease in students' feelings of management. The Pre-CPAQ results show a high degree of absence of perceived individual authority, as shown in Table 2 (Median=3.25, IQR=1.04), with a large proportion falling below the median and the median decreasing and the interquartile range increasing. Regarding using computers, there is a preponderance of negative (fearful) cognitions (Median=2.70, IQR=2.25), and the Post-CPAQ results show an increase in the negative cognitions.

C. Computer Self-Concept

Eleven items/statements were used to assess two aspects of computer self-concept: positive and negative. Students' answers to questions about their confidence or sense of self-efficacy were evaluated. Before the semester began in earnest, 18.5% of students in the Pre-CPAQ expressed doubt about their ability to help others to solve computer problems. Once more, when asked if they found it difficult to use computers, 15.3% of respondents said they strongly agreed.

Page : 445-452
Doi. https://doi.org/10.54209/jurnalinstall.v16i03.284

For a sizable portion of the student sample, the Pre-CPAQ clearly demonstrates poor positive computer self-concept, and there is little evidence of negative self-concept. Positive self-concept declined in Post-CPAQ analysis, and the findings indicate a fractional increase in negative self-concept. Thus, it is possible to draw the conclusion that when students acquire new skills and are exposed to difficult or novel material, they experience a period of low confidence and self-worth in the computing environment.

D. State of Anxiety in Computing Situations

This section of the survey looked at the cognitive, emotional, and physical symptoms of anxiety that students might experience when using computers. The findings represent each of the twenty-three statements' ratings for the four factors—happiness, worry, physical symptoms, and distractibility—that together indicate a person's anxiety level. The findings showed that while using or considering using a computer, a large number of students were content and very few felt anxious. Nonetheless, 47.4% of pupils reported feeling worried about possibles that might be happened when they were using computers. Compared to the other factors, physiological symptoms were less obvious, and 50% of students said that they were somewhat lack of concentration when using computers.

According to the Post-CPAQ, students' level of anxiety decreases over time. As Table 2 illustrates, the emotional state of happiness does not rise as one might anticipate. With reductions in the interquartile ranges, the physiological symptom and distractibility medians are the same in the pre- and post-analysis. The Post-CPAQ being distributed too soon before students' end-of-term exams may have contributed to the decline in happiness.

The results of the study show that while students' positive sense of control (cognition) does not improve in postanalysis, their positive self-concept does. When it comes to computers, students' anxiety does not go down; instead, it gets worse and their happiness goes down. By the end of the first year, students exhibit an unexpected increase in their negative self-concept, fear, and negative sense of control or self-talk.

**Table 2.** Pre-CPAQ and Post-CPAQ Results

| | PRE | | POST | |
|---|---|---|---|---|
| | Median | IQR | Median | IQR |
| Gaining Initial Computing Skills | | | | |
| Competence with Computers | 2.71 | 1.00 | 2.52 | 1.30 |
| Handling Computer Equipment | 2.50 | 1.50 | 2.24 | 1.62 |
| Receiving Feedback on Computing Skills | 2.60 | 1.20 | 2.32 | 1.50 |
| Learning about Basic Computer Functions | 2.59 | 1.04 | 2.20 | 1.20 |
| Sense of Control | | | | |
| Positive | 3.25 | 1.04 | 2.70 | 2.25 |
| Negative | 2.83 | 0.83 | 2.62 | 1.00 |
| Computer Self Concept | | | | |
| Positive | 2.75 | 0.75 | 2.60 | 1.20 |
| Negative | 3.80 | 1.60 | 3.52 | 1.70 |

Page : 445-452
Doi. https://doi.org/10.54209/jurnalinstall.v16i03.284

| | | | | |
|---|---|---|---|---|
| State of Anxiety in Computer Situations | | | | |
| Worry | 2.14 | 1.00 | 2.20 | 1.20 |
| Happiness | 3.16 | 0.88 | 2.75 | 0.89 |
| Physiological Symptoms | 2.00 | 1.20 | 2.00 | 1.00 |
| Distractibility | 2.50 | 1.00 | 2.50 | 0.50 |

## CONCLUSION

The degree to which people believe they can learn programming and have basic computer experience in comparison to their overall sense of computer proficiency is a significant determinant of their anxiety level. Even though the majority of students considered their level of computer experience to be "advanced," their responses to the Post-CPAQ at the end of the semester showed that they had reevaluated this term, with many saying that their perceived level of competence was much lower than they had previously thought. This brings up the subject of subjectivity in self-rating because it is possible for even very experienced students to feel inadequate and therefore see themselves as novices.

How well undergraduate students learn computer programming is significantly impacted by the data from the CPAQ's "state of anxiety in computing situation" scale. This study's low level of computing self-efficacy and degree of unfavourable beliefs about control in computing scenarios suggest that instructors of computer programming courses need to find ways to increase their students' confidence and motivation. This type of motivation will come from their insights of self-confidence, or the belief in the ability to learn and demonstrate programming proficiency in computer-related tasks. According to the results of "gaining initial computing skills," giving students the chance to practise skills necessary for regularly programming computerised equipment—like configuring programming environments or handling error messages—will help them become more confident and change their perspectives about programming course. The substantial degree of negative cognition expressed in the pre- and post-computer programming anxiety questionnaires in regard to views of control and computing self-confidence provides support to the concept which these fields of thinking require significant consideration.

## REFERENCES

[1] Alturki, R. A. (2016). "Measuring and improving student performance in an introductory programming course". Informatics in Education, 15(2), 183–204. https://doi.org/10.15388/infedu.2016.10
[2] Nikula, U., Gotel, O., & Kasurinen, J. (2011). "A motivation guided holistic rehabilitation of the first programming course". ACM Transactions on Computing Education, 11(4).
[3] Malcolm, C., Teague, D., & Thomas, R. N. (2010). "Engaging Students in Programming". 12th Australasian Computing Education Conference (ACE 2010), 63–72.
[4] Baist, A., & Pamungkas, A. S. (2017). "Analysis of Student Difficulties in Computer Programming". VOLT : Jurnal Ilmiah Pendidikan Teknik Elektro, 2(2), 81. https://doi.org/10.30870/volt.v2i2.2211
[5] Umar, I. N., & Hui, T. H. (2012). "Learning Style, Metaphor and Pair Programming: Do they Influence Performance?" Procedia - Social and Behavioral Sciences, 46, 5603–5609. https://doi.org/10.1016/j.sbspro.2012.06.482
[6] Radošević, D., Orehovački, T., & Lovrenčić, A. (2009). "Verificator: Mokomoji priemonė programavimui mokytis". Informatics in Education, 8(2), 261–280.

Page : 445-452
Doi. https://doi.org/10.54209/jurnalinstall.v16i03.284

[7] Owolabi, J., Olanipekun, P., & Iwerima, J. (2014). "Mathematics Ability and Anxiety, Computer and Programming Anxieties, Age and Gender as Determinants of Achievement in Basic Programming". GSTF Journal on Computing (JoC), 3(4), 109–114. https://doi.org/10.7603/s40601-013-0047-4
[8] Ashcraft, M. H., & Kirk, E. P. (2001). "The relationships among working memory, math anxiety, and performance". Journal of Experimental Psychology: General, 130(2), 224–237. https://doi.org/10.1037/0096-3445.130.2.224
[9] Acelejado, M. J. (2003). "The impact of using Technology on students' achievement, attitude and anxiety in Mathematics".
[10] McInerney, V. (1997). Computer Anxiety: "Assessment and Treatment". The University of Western Sydney Macarthur. https://doi.org/10.1145/2048931.2048935
[11] Marcoulides, G. A. (1988). "The Relationship between Computer Anxiety and Computer Achievement". Journal of Educational Computing Research, 4(2), 151–158. https://doi.org/10.2190/j5n4-24hk-567v-at6e
[12] Speier, C., Morris, M. G., & Briggs, C. M. (1995). "Attitudes Toward Computers: The Impact on Performance". Americas Conference on Information Systems.
[13] Mayer, R. E. (2009). "Multimedia Learning". Cambridge University Press.
[14] Pea, R. D., & Kurland, D. M. (1984). "On The Cognitive Effects of Learning Computer Programming". New Ideas Psychology, 2, 147–168.
[15] Connolly, C., Murphy, E., & Moore, S. (2009). "Programming Anxiety Amongst Computer Students-A Key in the Retention Debate?" IEEE Transactions on Education, 52(1), 52–56.
[16] Creswell, J. W. (2014). "Research Design: Qualitative, Quantitative, and Mixed Methods Approaches". In News.Ge. SAGE Publications.