

THE EFFECT OF VISUAL STUDIO CODE USABILITY BASED ON NIELSEN'S ATTRIBUTES ON THE PROGRAMMING PRODUCTIVITY OF INFORMATICS STUDENTS

Nadya Nada Pradista, Suprih Widodo

Pendidikan Sistem dan Teknologi Informasi, Universitas Pendidikan Indonesia, Purwakarta
supri@upi.edu

Article History

Received: 13 April 2026, Accepted: 28 April 2026, Published: 10 May 2026

Abstrak

Produktivitas pemrograman mahasiswa informatika dalam penyusunan tugas akhir sering menghadapi berbagai kendala, seperti kompleksitas kode, kesalahan teknis, serta keterbatasan dalam memahami dan memanfaatkan fitur tools pemrograman secara optimal. Kondisi tersebut menunjukkan bahwa kualitas usability dari code editor menjadi faktor penting yang dapat memengaruhi efektivitas, efisiensi, dan konsistensi kerja mahasiswa. Visual Studio Code sebagai salah satu code editor yang banyak digunakan memiliki berbagai fitur yang berpotensi mendukung produktivitas, namun pemanfaatannya masih bergantung pada persepsi dan pengalaman pengguna. Penelitian ini menggunakan pendekatan kuantitatif dengan metode survei untuk menganalisis pengaruh usability berdasarkan Nielsen's Attributes of Usability terhadap produktivitas pemrograman mahasiswa. Data dikumpulkan melalui kuesioner yang mengukur dimensi learnability, efficiency, memorability, errors, dan satisfaction. Analisis dilakukan menggunakan regresi linear berganda. Hasil penelitian menunjukkan bahwa seluruh dimensi usability berpengaruh positif dan signifikan terhadap produktivitas pemrograman, dengan efficiency sebagai dimensi yang paling dominan. Temuan ini menunjukkan bahwa kualitas usability tools pemrograman berperan penting dalam meningkatkan kinerja mahasiswa.

Kata Kunci: usability; VS Code; NAU; produktivitas pemrograman

Abstract

Programming productivity among informatics students in completing final projects often encounters various challenges, including code complexity, technical errors, and limited ability to fully utilize programming tools. These conditions indicate that the usability quality of code editors plays a crucial role in influencing students' effectiveness, efficiency, and consistency in programming activities. Visual Studio Code, as a widely used code editor, provides various features that potentially support productivity, yet its effectiveness depends on users' experience and perception. This study employs a quantitative approach with a survey method to examine the effect of usability based on Nielsen's Attributes of Usability on programming productivity. Data were collected using a structured questionnaire measuring learnability, efficiency, memorability, errors, and satisfaction. The analysis was conducted using multiple linear regression. The findings reveal that all usability dimensions have a positive and significant effect on programming productivity, with efficiency identified as the most dominant dimension. These results indicate that the usability quality of programming tools plays an important role in improving students' performance.

Keyword: usability, VS Code, NAU, programming productivity

To cite this article:

Pradista, N. N., & Widodo, S. (2026). The effect of visual studio code usability based on nielsen's attributes on the programming productivity of informatics students. *JKTP: Jurnal Kajian Teknologi Pendidikan*, 9(2), 244-253. doi: [10.17977/um038v9i22026p244-253](https://doi.org/10.17977/um038v9i22026p244-253)

INTRODUCTION

Programming productivity is one of the key aspects that determines the success of Informatics students in completing their final projects. The preparation of software development-based final projects requires students to write, test, and debug program code intensively within a limited period of time. These activities demand not only strong conceptual understanding, but also adequate technical skills and the ability to manage time effectively (Abdurrokhim, Kuswandi, and Ulfa 2022). The ability to maintain consistency throughout the coding process also becomes a challenge, particularly when students face increasingly complex systems. This condition requires students to work effectively and efficiently in order to complete projects on time with satisfactory quality. Issues related to programming productivity have become increasingly relevant along with the growing complexity of final project development, which now involves various modern technologies and frameworks (Halim, 2025).

The challenges encountered during the preparation of final projects are not only conceptual, but also technical. Students frequently experience difficulties in managing increasingly complex code, performing time-consuming debugging processes, and adapting to various libraries and frameworks (Adi and Wedi 2021). Minor errors in code writing can significantly affect the entire system and therefore require repeated checking processes. A lack of experience in handling specific errors often causes project completion to take longer (Rukmi, Sulton, and Husna 2020). Deadline pressure and simultaneous academic demands further intensify these conditions. A study conducted by Baltes and Diehl (2016) showed that technical obstacles during the coding process significantly contribute to delays in software development project completion, including within academic contexts.

Selecting appropriate programming tools becomes an important strategy in supporting student productivity. Integrated Development Environments (IDEs) and code editors act as the primary instruments used daily for writing and managing program code. Tools with comprehensive features and user-friendly interfaces can help students work in a more structured and efficient manner. In contrast, tools that are difficult to understand may hinder workflow and reduce efficiency. The quality and usability of programming tools directly affect speed, accuracy, and comfort during work processes. Usability aspects therefore become important factors shaping user experience while simultaneously influencing programming outcomes. Positive user experiences can increase motivation, whereas suboptimal experiences may reduce work performance (Murani et al. (2025)).

Visual Studio Code is a code editor developed by Microsoft and has become one of the most popular programming tools among developers and students. This popularity reflects users' trust in its performance and available features. Based on the Stack Overflow Developer Survey 2024, VS Code ranked as the most widely used code editor globally. Various advanced features such as IntelliSense for smart autocomplete, integrated terminal, flexible extension systems, integrated debugging support, and integration with version control systems such as Git are designed to improve work efficiency. Its relatively simple interface also allows new users to adapt more easily without reducing the needs of experienced users (Arwani, Az-zahra, and Rokhmawati 2024).

The concept of usability is used as a theoretical framework to evaluate the extent to which a tool can support its users. Usability is not only related to ease of use, but also encompasses effectiveness and user satisfaction during interactions with the system. Jakob Nielsen introduced five main attributes known as Nielsen's Attributes of Usability (NAU), consisting of learnability, efficiency, memorability, errors, and satisfaction. Learnability relates to how easily new users can understand the system, efficiency describes how quickly experienced users can complete tasks, memorability refers to the ease of remembering how to use the system, errors relate to the frequency of mistakes and recovery processes, and satisfaction reflects the overall level of user satisfaction.

These five dimensions form a comprehensive usability construct that is widely used in evaluating interactive systems (Oktaroza and Setiawan (2025)).

The relationship between usability and productivity has become a major focus in various software development studies. Tools with high usability levels enable users to work faster and more accurately because they do not require excessive time to understand the system. Users can focus directly on task completion without being distracted by unnecessary technical barriers. Efficiency aspects such as shortcuts and autocomplete have been proven to accelerate the coding process, while the errors aspect helps users identify and correct mistakes more effectively. User comfort also contributes to maintaining concentration and work stability throughout programming activities (Tjahyanti, Saputra, and Santo Gitakarma 2022).

Research related to IDE usability still shown several limitations, particularly in empirically linking usability aspects with productivity. Many previous studies have focused solely on usability evaluation without measuring its impact on user performance. The use of the NAU framework specifically for code editors such as VS Code also remains relatively limited. Most previous studies have concentrated more on web-based applications, e-learning systems, or organizational information systems. This condition indicates a relevant research gap that deserves further investigation, especially in the context of IDE usage among Informatics students (Supiyan 2026).

Based on the background described above, this study is directed toward analyzing the influence of VS Code usability dimensions based on Nielsen's Attributes of Usability on the programming productivity of Informatics students during final project preparation. The analysis is conducted by examining the effects of each dimension, namely learnability, efficiency, memorability, errors, and satisfaction, on programming productivity both partially and simultaneously. This approach enables the identification of the most dominant dimensions, thereby providing a clearer understanding of the factors contributing to improved programming productivity. The findings are expected to strengthen understanding regarding the role of usability in supporting students' programming performance and serve as a basis for formulating practical recommendations for students and educational institutions in optimizing the use of programming tools, in line with the research focus entitled "The Effect of Visual Studio Code Usability Based on Nielsen's Attributes on the Programming Productivity of Informatics Students."

METHOD

This study employed a quantitative approach using a survey method to measure the influence of VS Code usability dimensions on students' programming productivity. The research applied an *ex post facto* design, in which variables were measured based on existing conditions without providing any specific treatment. This study was conducted in Informatics study programs at higher education institutions whose students used Visual Studio Code as the primary programming tool during the preparation of their final projects.

The research population consisted of all active Informatics students who were currently undertaking or had completed the final project process using VS Code as their primary programming tool. The sample comprised 200 students selected through purposive sampling techniques, with criteria including active Informatics students in the seventh semester or above, students currently working on or having completed their final projects, and students using VS Code as their main code editor during programming activities. Based on respondent characteristics, the majority of respondents were male students, with 118 respondents (59%), while female students consisted of 82 respondents (41%). Semester distribution showed that most respondents were from the eighth semester with 80 students, followed by the seventh semester with 72 students, the ninth semester with 34 students, and the tenth semester with 14 students.

The study consisted of five independent variables and one dependent variable. The independent variables included learnability (X1), which measured users ability to understand the VS Code interface and features; efficiency (X2), which measured how quickly users completing coding tasks; memorability (X3), which measured how easily users could recall features after periods of non-use; errors (X4), which measured the low occurrence of operational errors and the ease of resolving them; and satisfaction (X5), which measured users' comfort and satisfaction with VS Code. The dependent variable was programming productivity (Y), which included coding performances, time efficiency, work consistency, and final project progress.

The research instrument consisted of a closed-ended questionnaire using a 5-point Likert scale with a total of 26 statement items: four items for each independent variable and six items for the dependent variable. Data were collected online through a digital questionnaire distributed to respondents who met the specified criteria. Data analysis employed multiple linear regression to examine both partial and simultaneous effects. Prior to regression analysis, prerequisite tests were conducted, including validity testing, reliability testing, normality testing, multicollinearity testing, and heteroscedasticity testing to ensure the suitability of the regression model.

RESULT

The respondent characteristics indicated that most students were actively engaged in the final project preparation phase, with male students dominating the sample at 118 respondents (59%), while female students accounted for 82 respondents (41%). In terms of semester distribution, most respondents were in the seventh semester (36%) and eighth semester (40%), followed by the ninth semester (17%) and tenth semester (7%). This distribution demonstrates strong relevance to the context of the study, particularly the use of programming tools during final project development, as reflected in the descriptive analysis of the research variables presented in Table 1.

Table 1. Descriptive Statistics of Research Variables (N = 200)

Variable	Mean	Std. Dev	Min	Max
X1 (Learnability)	3,896	0,678	2,00	5,00
X2 (Efficiency)	3,621	0,746	1,50	5,00
X3 (Memorability)	3,643	0,713	1,50	5,00
X4 (Errors)	3,523	0,770	1,25	5,00
X5 (Satisfaction)	3,870	0,681	2,00	5,00
Y (Productivity)	3,874	0,510	2,33	5,00

Based on Table 1, all usability dimensions showed mean scores that generally fell within the high category (above 3.50), although the errors variable was positioned at the lower boundary of that category. The learnability variable (M = 3.896) and satisfaction variable (M = 3.870) demonstrated relatively higher mean scores compared to the other dimensions, indicating that Visual Studio Code was perceived as easy to learn and capable of providing a sufficiently satisfying user experience. In contrast, the errors variable recorded the lowest mean score (M = 3.523), suggesting that error handling remained an aspect requiring further attention. Meanwhile, programming productivity obtained a mean score of 3.874 with a relatively small standard deviation (0.510), indicating that respondents' perceptions of productivity tended to be consistent.

The researcher first examined the quality of the instrument to ensure that the collected data could be used for further analysis. The validity test showed that all items met the required criteria because the Corrected Item–Total Correlation values exceeded the r-table value (0.1388), ranging from 0.491 to 0.740, indicating that each item was valid. Furthermore, the reliability test demonstrated that all variables had Cronbach's Alpha values above 0.70, namely learnability (0.850), efficiency (0.858), memorability (0.849), errors (0.863), satisfaction (0.850), and

productivity (0.799). These results indicate that the research instrument was able to measure the variables consistently.

After confirming that the instrument was appropriate, the researcher proceeded with classical assumption testing to determine whether the regression model was suitable for analysis. In the normality test, the Kolmogorov–Smirnov method produced a significance value of 0.858, which exceeded 0.05, indicating that the residuals were normally distributed. All independent variables were also tested for multicollinearity, and the results showed Tolerance values above 0.10 and VIF values below 10 (ranging from 1.018 to 1.077), indicating that no multicollinearity issues were identified among the variables. The heteroscedasticity test using the Glejser method also produced satisfactory results, as the significance values ranged from 0.373 to 0.815, all exceeding 0.05, which indicates homogeneous residual variance. Since all assumptions were fulfilled, the regression model was considered suitable for hypothesis testing. The results of the multiple linear regression analysis are presented in Table 2.

Table 2. Results of Multiple Linear Regression Analysis

Variabel	B	Std. Error	t	Sig.
(Constant)	0,139	0,232	0,598	0,551
Learnability (X1)	0,205	0,034	5,990	0,000
Efficiency (X2)	0,301	0,032	9,377	0,000
Memorability (X3)	0,176	0,033	5,403	0,000
Errors (X4)	0,149	0,031	4,798	0,000
Satisfaction (X5)	0,176	0,035	5,093	0,000

$R^2 = 0,605$; $Adjusted R^2 = 0,595$; $F = 59,357$ (Sig. = 0,000)

In Table 2, the multiple linear regression equation is expressed as $Y = 0.139 + 0.205X1 + 0.301X2 + 0.176X3 + 0.149X4 + 0.176X5 + e$. This equation indicates that all usability dimensions of VS Code had a positive relationship with students' programming productivity. In other words, when students perceived VS Code as easier to use, their productivity in completing final projects also increased. The constant value of 0.139 indicates that even without the influence of usability variables, a certain level of productivity would still exist, although at a lower level.

The partial testing results showed that all independent variables had a significant influence on programming productivity. The efficiency variable emerged as the most dominant factor, with a regression coefficient of 0.301, a t-value of 9.377, and a significance value of 0.000. This indicates that the more efficiently students used VS Code features such as autocomplete, shortcuts, extension integration, and navigation, the faster and more effective their programming process became. In addition, learnability also contributed substantially, with a coefficient value of 0.205. Meanwhile, satisfaction and memorability recorded identical coefficient values of 0.176, while errors had the smallest coefficient value at 0.149. These findings indicate that all usability dimensions contributed to improving programming productivity, although their levels of influence varied.

The simultaneous test results further strengthened these findings. The regression model was declared significant because the calculated F-value was 59.357 with a significance value of 0.000. This means that all usability dimensions simultaneously influenced students' programming productivity. The magnitude of this influence can be observed from the coefficient of determination value of 0.605, indicating that a substantial proportion of productivity variation could be explained by the variables of learnability, efficiency, memorability, errors, and satisfaction. The remaining variation was influenced by factors outside the model, such as technical experience, programming language proficiency, time management, and the level of difficulty of the students' final projects.

DISCUSSION

This study demonstrated that all usability dimensions of VS Code based on Nielsen's Attributes of Usability (NAU) had a positive and significant influence on the programming productivity of Informatics students. These findings indicate that the quality of interaction between students and the programming tool was well supported, enabling the final project development process to be carried out more effectively. The coefficient of determination value of 0.605 further indicated that the variation in programming productivity could be explained quite strongly by the research model. Therefore, usability should not merely be viewed as a matter of user convenience, but also as a factor that contributes to work effectiveness and the quality of programming outcomes.

Conceptually, these findings can be explained through the NAU framework, in which learnability, efficiency, memorability, errors, and satisfaction are positioned as dimensions that shape the quality of interaction between users and systems. Within this framework, usability is understood as a multidimensional construct that collectively determines the level of system-use productivity. This perspective is consistent with the ISO 9241-11 standard, which defines usability in terms of a product's ability to be used effectively, efficiently, and satisfactorily in achieving specific goals (Artamevia and Paputungan (2025)). In the context of programming-based final projects, VS Code functioned as the primary tool whose usability quality directly influenced students' effectiveness in achieving academic objectives. The R^2 value of 0.605 indicates that Nielsen's Attributes of Usability framework was sufficiently capable of explaining variations in programming productivity.

These findings are also supported by previous studies. Usability has been identified as a factor capable of improving effectiveness, efficiency, and user satisfaction in task completion (Sopandi and Wahyu 2024; Wachnata, Jandhana, and Aritonang (2025)). Furthermore, studies on interface design have shown that the implementation of usability principles can improve task-completion efficiency and reduce user error rates, thereby contributing to increased productivity (Amesha et al. (2025); Rustiawan 2026). In broader studies, software developer productivity has also been influenced by the quality of the tools used during the development process (Cheng et al. 2022).

In addition, these findings can also be interpreted through the SPACE framework proposed by Forsgren et al. (2021), which includes dimensions such as satisfaction, performance, activity, communication, efficiency, and flow. The framework emphasizes that software developer productivity should not be assessed solely based on the quantity of code or commits produced, but rather through overall software development performance, including user satisfaction, workflow continuity, and effectiveness in task completion. The efficiency and satisfaction dimensions in NAU conceptually align with the Efficiency and Flow as well as Satisfaction and Well-being dimensions in the SPACE framework. Therefore, the finding that these two dimensions significantly contributed to students' programming productivity is consistent with modern theoretical frameworks of developer productivity (Anshori (2025)). Accordingly, the usability quality of VS Code functions not only as a technical variable, but also as a predictor of developer experience that is highly relevant within academic contexts.

The dominance of the efficiency dimension indicated that it contributed the most to students' programming productivity. This finding suggests that efficient use of VS Code was directly associated with students' ability to complete coding tasks more quickly and systematically (Abdillah, Siregar, and Ramadhany (2025)). Features such as keyboard shortcuts, IntelliSense, autocomplete, and extension integration helped accelerate the coding process while minimizing technical obstacles. This condition demonstrates that efficiency is not solely related to work speed,

but also to workflow continuity throughout the programming process (Alida, Thyas, and Jajuli (2025)).

These findings are consistent with the study conducted by Hliš et al. (2023), which reported that code-completion features were capable of increasing coding speed and reducing errors, thereby contributing to improved developer productivity. Research by Murani et al. ((2025)) also showed that the use of Visual Studio Code enhanced students' efficiency and coding skills through features that practically supported programming activities. The dominant influence of efficiency indicates that students tended to rely heavily on features that provided direct impacts on work speed, particularly during final project development, where effective time management was essential (Alda et al. (2025)).

The learnability dimension demonstrated that the ease of understanding and mastering VS Code features also supported programming productivity. Students who adapted more quickly to the IDE environment had more time to focus on task completion (David, Rindengan, and Diva 2026). This finding aligns with the concept of learnability in usability theory, which states that systems that are easier to learn reduce the cognitive burden associated with understanding how the system operates (Oktaroza and Setiawan (2025)). Research by Murani et al. ((2025)) further showed that intuitive interfaces helped students improve their understanding and programming skills.

The memorability dimension also influenced students' programming productivity. The ability to easily recall how to use the system after periods of non-use allowed students to resume their work without having to relearn the system from the beginning. This condition is particularly relevant to the nature of final project work, which is often carried out intermittently rather than continuously. Consistent interfaces and simple usage patterns helped maintain students' workflow continuity (Chyan et al. (2025)).

The errors dimension showed a significant influence, although its contribution was lower compared to the other dimensions. VS Code's ability to assist users in understanding and resolving errors still played an important role in supporting productivity. Features such as error notifications, debugging tools, and undo functionality helped students identify and fix bugs more efficiently. These findings are consistent with Rustiawana's study, which demonstrated that systems with low error rates and effective error-handling support could improve the effectiveness of user interaction (Syafei and Hidayatullah 2023). However, the relatively lower contribution of this dimension indicates that students had not yet fully optimized the debugging features available in VS Code.

The satisfaction dimension indicated that a comfortable user experience contributed positively to programming productivity. An intuitive interface, flexible customization options, and user-friendly features helped students maintain focus during coding activities. This finding aligns with the study conducted by Rachmawati (2025), which stated that user experience is closely related to productivity because it affects user comfort and engagement during work activities. Research by Murani et al. (2025) also showed that high levels of satisfaction with VS Code usage were associated with improved student performance in programming.

The synthesis of these five dimensions demonstrates that the relationship between usability and productivity can be positioned within two complementary theoretical perspectives. First, from the Human-Computer Interaction (HCI) perspective, usability quality shapes a positive user experience, enabling students to interact with the IDE intuitively and comfortably (Stefanidi et al., 2024). Second, from the software engineering productivity perspective, usability dimensions function as enablers of Efficiency and Flow as well as Satisfaction within the SPACE framework, both of which are considered strong predictors of developer productivity (Forsgren et al. 2021). The convergence

of these two perspectives strengthens the argument that VS Code functions not merely as a coding tool, but also as a learning environment that shapes students' programming-learning experiences. Consequently, improving students' programming productivity cannot be achieved solely through mastery of programming languages, but also through the comprehensive optimization of IDE usability features.

Overall, the findings of this study indicate that students' programming productivity was influenced by the overall quality of usability. Each usability dimension contributed complementarily in shaping a user experience that supported workflow processes. These findings reinforce the relevance of Nielsen's Attributes of Usability in explaining the relationship between programming-tool quality and user productivity within Informatics education contexts.

CONCLUSION

This study showed that the usability of VS Code based on the NAU framework, which includes learnability, efficiency, memorability, errors, and satisfaction, had a positive and significant influence on the programming productivity of Informatics students. Among these dimensions, efficiency emerged as the most dominant factor. The coefficient of determination value of 0.605 indicated that the model was able to explain variations in productivity adequately, suggesting that the usability quality of a code editor can be regarded as one of the factors influencing students' work effectiveness and efficiency during the completion of their final projects. The findings also demonstrated that the NAU framework is relevant for explaining the relationship between the usability quality of VS Code and programming productivity within an academic context. However, these findings should still be interpreted by considering several limitations of the study. The survey-based approach relying on user perceptions has not fully represented actual usage behavior, while the respondents being limited to Informatics students means that the findings cannot yet be generalized more broadly. In addition, the analysis was primarily focused on usability aspects, so other factors such as technical experience, task complexity, and time management were not examined. Future studies are therefore recommended to apply more in-depth approaches, such as experimental methods or direct observation, in order to better reflect actual conditions. Furthermore, expanding the respondent scope and incorporating other relevant variables are necessary to provide a more comprehensive understanding of the factors influencing programming productivity.

REFERENCES

- Abdillah, M. H., Siregar, H., & Ramadhany, F. (2025). Korelasi Kemampuan Coding dengan Minat Berwirausaha Teknologi di Kalangan Mahasiswa Ilmu Komputer. *Jurnal Pendidikan Sains dan Komputer*, 5(02), 333-343.
- Abdurrokhim, A., Kuswandi, D., & Ulfa, S. (2022). Pengembangan pembelajaran berbasis web dengan pendekatan guided discovery berbantuan hypermedia untuk siswa SMP. *JKTP: Jurnal Kajian Teknologi Pendidikan*, 5(2), 121-131.
- Adi, Eka Pramono, and Wedi, A. (2021). "Studi Analisis Prokrastinasi Akademik Mahasiswa Dalam Mengerjakan Skripsi." *JKTP: Jurnal Kajian Teknologi Pendidikan* 4(4):358–67.
- Alda, M., Febryani, N., Safitri, P., & Albar, M. (2025). Pengaruh Dampak Penggunaan Aplikasi Mobile Terhadap Tingkat Produktivitas Mahasiswa. *Jurnal RESTIKOM: Riset Teknik Informatika dan Komputer*, 7(3), 498-505.
- Alida, M., Thyas, L. A. K., & Jajuli, A. R. (2025). Implementasi Website Weji Coffee dengan Fitur Reservasi Menggunakan HTML, JavaScript, dan Tailwind CSS. *Jurnal Minfo Polgan*, 14(2), 3033-3042.

- Amesha, S. A., Adinda, T., Nazira, N., & Willyansah, W. (2025). Pengaruh desain ui/ux yang efektif dalam meningkatkan pengalaman dan kepuasan penggunaan. *Jurnal Data Sains Dan Teknologi Informasi (DASTIS)*, 2(02), 1-10.
- Anshori, Mohamad Yusak. (2025). *Softbrain Engineer Membangun Kecerdasan Organisasi Di Era Disrupts*. Deepublish.
- Artamevia, Kharisma Suci, and Irving Vitra Paputungan. (2025). "Evaluasi Usability Aplikasi Mobile Peminjaman Fasilitas Olahraga Menggunakan Metode ISO 9241-11." *Technologia: Jurnal Ilmiah* 16(4):696–703.
- Arwani, W. N. M., Az-zahra, H. M., & Rokhmawati, R. I. (2024). Perbedaan Typefaces Serif Dengan Sans Serif Terhadap Pengalaman Membaca Modul Desain Interaksi Media Pembelajaran. *JKTP: Jurnal Kajian Teknologi Pendidikan*, 7(2), 058-067.
- Baltes, Sebastian, and Stephan Diehl. (2016). "Worse than Spam: Issues in Sampling Software Developers." Pp. 1–6 in *Proceedings of the 10th ACM/IEEE international symposium on empirical software engineering and measurement*.
- Cheng, Lan, Emerson Murphy-Hill, Mark Canning, Ciera Jaspan, Collin Green, Andrea Knight, Nan Zhang, and Elizabeth Kammer. (2022). "What Improves Developer Productivity at Google? Code Quality." Pp. 1302–13 in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*.
- Chyan, Phie, Amril Mutoi Siregar, Ahmad Afif Mauludi, Novita Lestari Anggreini, Ridnaldy Yuniar Carolus, Agung Yuliyanto Nugroho, Zulfiana Safitri Majid, Ely Aprilia, M. Rhifky Wayahdi, and Arif Iman Anshori. (2025). *Interaksi Manusia Dan Komputer: Membangun Antarmuka Cerdas Di Era Digital*. Mifandi Mandiri Digital.
- David, Jeremia, Yaulie Deo Rindengan, and Fiorentine Diva. (2026). *Membangun Aplikasi Pembelajaran Interaktif Mobile Learning: Strategi Agile Dari Konsep Hingga Development*. Minhaj Pustaka.
- Forsgren, Nicole, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler. (2021). "The SPACE of Developer Productivity: There's More to It than You Think." *Queue* 19(1):20–48.
- Halim, Amar. (2025). "Merancang Kurikulum Deep Learning Dan Coding Berbasis Kebutuhan Industri Teknologi Informasi." *JIMU: Jurnal Ilmiah Multidisipliner* 3(04):2312–25.
- Hliš, Tilen, Luka Četina, Tina Beranič, and Luka Pavlič. (2023). "Evaluating the Usability and Functionality of Intelligent Source Code Completion Assistants: A Comprehensive Review." *Applied Sciences* 13(24):13061.
- Murani, Nia Ani, Sri Wahyuni, Rindi Ambar Pratiwi, and Thoha Firdaus. (2025). "Analisis Penggunaan Visual Studio Code Sebagai Editor Pemrograman Terbaik Untuk Pemula Pada Mahasiswa Informatika Universitas Nurul Huda." *Jurnal Rekayasa Informatika* 2(2):1–10.
- Oktaroza, Magnaz Lestira, and Amelia Setiawan. (2025). "Efektivitas Dan Usability User Interface Dalam Aplikasi Modern: Tinjauan Sistematis Melalui Studi Literatur Review." *Jurnal Manajemen Terapan Dan Keuangan* 14(02):475–86.
- Rachmawati, Adinda Putri. (2025). "Evaluasi Usability Sistem Informasi Menggunakan Metode User Experience Questionnaire." *Jurnal Sistem Informasi* 1(1):9–15.
- Rukmi, J. N. P., Sulton, S., & Husna, A. (2020). Efektivitas model pembelajaran team game tournament (TGT) siswa sekolah menengah atas pada materi sistem koordinasi. *JKTP: Jurnal Kajian Teknologi Pendidikan*, 3(4), 456-465.
- Rustiawan, Andry. (2026). "Analisis Pengaruh Desain UI Terhadap Efektivitas Interaksi Pengguna." *Jurnal Komputer Dan Teknologi Informasi (JUKOMTIK)* 1(2):80–86.

- Sopandi, Ajang, and Angelica Wahyu. (2024). “Evaluasi Usability Tableau Desktop Menggunakan Metode Nielsen’s Attributes of Usability (NAU).” *Jurnal Perangkat Lunak* 6(2):287–95.
- Supiyani, Dede. (2026). *Dasar–Dasar Pemrograman Berbasis Processing*. CV Eureka Media Aksara.
- Syafei, Tubagus Fandi Maulana, and Azka Hidayatullah. (2023). “Analisis Penerapan UI/UX Dalam Meningkatkan Pengalaman Pengguna Pada Sistem Reservasi Amadeus.” *Jurnal Sistem Informasi Dan Teknologi Informasi* 1(1):1–8.
- Tjahyanti, Luh Putu Ary Sri, Putu Satya Saputra, and Made Santo Gitakarma. (2022). “Peran Artificial Intelligence (AI) Untuk Mendukung Pembelajaran Di Masa Pandemi Covid-19.” *KOMTEKS* 1(1).
- Wachnata, Dascha Abhista, I. B. Putra Jandhana, and Sovian Aritonang. (2025). “Perkembangan Konsep User Experience Dalam Teknologi Pertahanan Modern: Studi Literatur.” *JATI (Jurnal Mahasiswa Teknik Informatika)* 9(5):7906–13.