SCIENCE AND INFORMATION TECHNOLOGY
SINTECH
JOURNAL

SINTECH JOURNAL | ISSN 2598-7305 | E-ISSN 2598-9642
Vol. 4 No 1 – April 2021 | https://s.id/sintechjournal
DOI : https://doi.org/10.31598
Publishing : LPPM STMIK STIKOM Indonesia

PROGRAMMER'S PERSPECTIVE OF OBJECT ORIENTED PROGRAMMING (OOP) IN SOFTWARE DEVELOPMENT USING CORRELATION ANALYSIS

Bagas Triaji[1], Cucut Hariz Pratomo[2], Bambang Purnomosidi DP[3]

[1,2,3] Master of Information Technology, STMIK AKAKOM Yogyakarta
Banguntapan, Bantul, Yogyakarta, Indonesia

Email: student.bagastriaji@mti.akakom.ac.id[1], student.cucuthariz@mti.akakom.ac.id[2], bpdp@akakom.ac.id[3]

Received : August, 2020	Accepted : December, 2020	Published : April, 2021

Abstrak

Pesatnya perkembangan teknologi menghasilkan era digitalisasi. Permintaan pengembangan perangkat lunak dan insinyur perangkat lunak di berbagai sektor industri, bisnis, dan pendidikan sangat tinggi. Yogyakarta adalah kota pendidikan, dimana banyak perguruan tinggi dan universitas berdiri. Namun, calon programmer sering memiliki pemahaman yang kurang memadai tentang paradigma OOP dari perspektif praktisi industri IT. Oleh karena itu, survei berikut melibatkan praktisi programmer profesional dilakukan untuk menganalisis bagaimana mereka melihat Object-Oriented Programming (OOP) ketika mengembangkan perangkat lunak dan bagaimana pengalaman mereka, dengan menggunakan analisis korelasi. Penelitian ini dilakukan untuk mengkaji aspek yang mempengaruhi preferensi programmer terhadap OOP. Hasil analisis korelasi menunjukkan bahwa programmer yang lebih berpengalaman akan lebih memilih paradigma OOP untuk menyelesaikan proyek meskipun mengalami beberapa hambatan dalam implementasi OOP, tetapi mereka tidak yakin bahwa OOP akan tetap digunakan sebagai paradigma yang mumpuni di masa depan.

Kata Kunci : *OOP, Object-Oriented Programming, Paradigma Pemrograman, Programmer, Analisis Korelasi.*

Abstract

The rapid development of technology emerges the era of digitalization. The demand for software development and software engineers in various industrial sectors, business, and education is very high. Yogyakarta is a city of education where many colleges and universities were established. However, prospective programmers often had inadequate understanding about OOP paradigm from the perspective of IT industry practitioners. Therefore, a survey involving professional programmer practitioners was carried out to analyze how they view Object-Oriented Programming (OOP) when developing software and how they use correlation analysis. This study was done to examine aspects affecting programmers' preferences. The results of correlation analysis indicated that more experienced programmers would rather choose the OOP paradigm to complete the project and experience a few obstacles in OOP implementation but they were not sure that OOP would remain as a strong paradigm in the future.

Keyword : *OOP, Object-Oriented Programming, Programming Paradigm, Programmer, Correlation Analysis.*

1. INTRODUCTION

The demand for software development in this digital era is increasing as various industrial sectors have started using software for efficiency purposes. The demand for complex software also keeps increasing that programmers need to apply different programming paradigms to solve this problem.

Programming paradigm is a way a programmer sees the programming software to solve certain problem [1]. There are several programming paradigms such as procedural, concurrent functional programming and object-oriented, event-driven.

An object-oriented paradigm is an approach that is mainly focused on how objects interact to communicate and share information. Object-oriented programming covers at least 3 goals of software engineering that are reusability, extensibility, and flexibility [2]. Object-oriented programming (OOP) is still used in developing mobile software ever since it was first introduced in 1960. [3]. However, a study done in Oslo, Norway showed some difficulties for students who learn OOP to perform a task using the framework of medium-large Java applications [4].

Prior surveys on object oriented software engineering among practitioners have been carried to find out techniques and tools in designing object-oriented software-based and examine the relationship between the confidence of experienced practitioners and the quality of software design [5].

In Indonesia, especially in Yogyakarta, object-oriented paradigm is the most popular paradigm. Both in universities and industries, OOP has been a favorite option because people believe it can accelerate the software development with lower cost [6] .

OOP is set as a compulsory subject in the curriculum of software development program in every university or college based in the *SKKNI (Standar Kompetensi Kerja Nasional Indonesia)* [7].

There have been 138 universities registered in Yogyakarta, 58 out of which have information technology program [8]. These universities produce skillful programmers to be recruited by digital startup companies which number has grown to 54 companies and ranked 3rd in Indonesia in 2018 [9].

Unfortunately, most prospective graduate programmers do not understand the real condition in the workplace, especially related to the use of OOP paradigm. Most of them had low motivation in to enhance their competence and they were lack of focus as they were confused by many paradigms. Therefore, insights from IT industry regarding the programming paradigm issue are needed.

In this paper, programmers' perspectives about the OOP paradigm were examined to obtain information about the relationship between OOP comprehension and its application. This research also analyzed the advantages and disadvantages of OOP compared in its implementation in software development.

2. METHOD

A correlation analysis was administered to data which were collected through questionnaires. Questionnaires were distributed online using Google Forms consisting of 5 parts:

(1) Respondents' profile

(2) Preference of programming paradigms. This session collected information about respondents' experiences and knowledge of programming paradigms.

(3) Experience in handling programming projects. At this stage, information about the OOP programming paradigm based on respondents' experience and project work in the institution were collected.

(4) Obstacles. Information on the problems of using OOP by respondents were collected.

(5) Recommendation. At this stage, respondents provided information about the OOP paradigm in the form of some recommendations.

The target respondents were programmers who worked in Yogyakarta with various specialties, such as web, mobile, and dekstop development. Respondents' programming skill was categorized into junior and senior level [10] . There were 30 programmers from startup companies and other companies including PT Koltiva, PT Gamatechno, PT Ainosi, PT Cipta Karya, PT Solusi Kampus Indonesia, and others participated in this study. Most of the companies were engaged in software development services.

The data of this study were analyzed using Spearman and Pearson test and One-Sample Kolmogorov-Smirnov Test implies that :

- If asymp.sig.(2-tailed) < α means the data is not normally distributed.
- Then if asymp.sig.(2-tailed) > α means the data is normally distributed.
  The significance level used (α) = 0,05

in SPSS program to find the correlation between the variables. Statistical tests were performed to determine the relationship between programmers' preference and experience in handling project management, programmer experience in dealing with obstacles and their recommendations.

Each variable shoed certain score which can affect programmers' preference and experience in dealing with programming projects, The programming experience related to the obstacles and respondents' recommendations had the strongest influence on programmers' preference.

3. RESULTS AND DISCUSSIONS

This section describes the data of this study and the results of correlation analysis.

3.1 Results

The data obtained from the questionnaires are presented in Table 1. Table 1 shows the weighting of the data from the questionnaire and the description of respondents as follows.

Table 1: Weight of Data

Respondent	Section. 2 ; Programmers' preferences of programming paradigm	Section 3 ; Experience of handling projects	Section 4 ; Experience of handling obstacles	Section 5 ; Conclusion
1	14	24	7	19
2	17	25	10	24
3	15	23	8	21
4	17	23	6	22
5	15	21	6	20
6	13	21	7	22
7	9	22	7	21
8	25	25	9	25
9	18	22	11	22
10	16	25	7	25
11	9	25	5	24
12	14	20	10	22
13	12	17	13	17
14	16	20	13	16
15	12	19	16	18
16	20	25	9	25
17	19	25	7	17
18	19	15	10	21
19	14	15	21	23
20	22	25	5	25
21	17	19	15	18
22	20	25	10	19
23	19	25	10	18

24	15	17	11	17
25	15	25	10	21
26	13	14	16	12
27	14	22	11	18
28	15	20	11	21
29	19	18	10	17
30	16	23	9	21

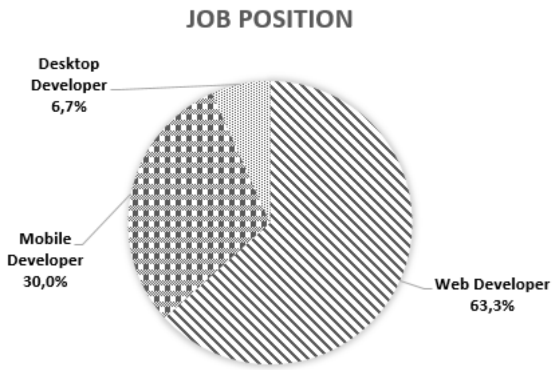

Fig. 1. Job Position

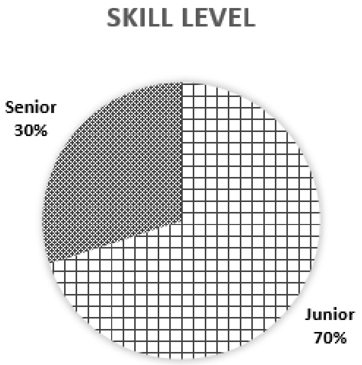

Fig. 2. Skill Level

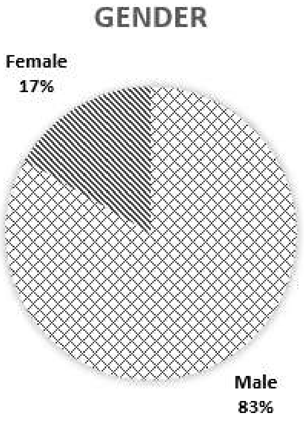

Fig. 3. Gender

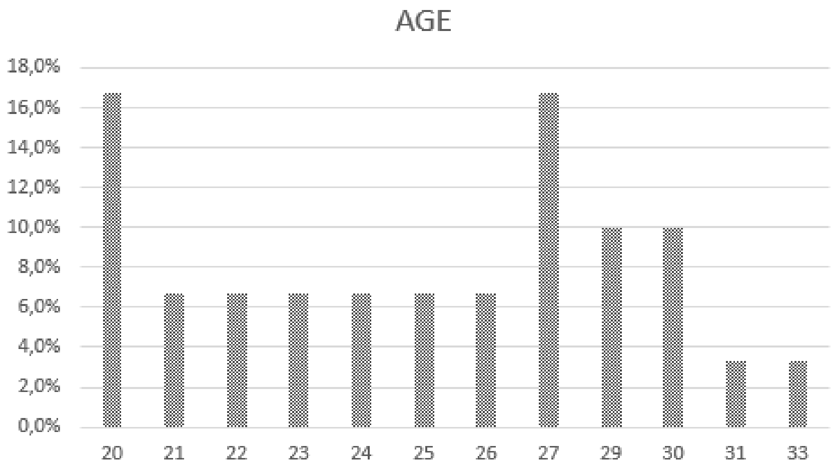

Fig. 4. Age

Fig1 shows the categorization of respondents' programming tasks into Web Developers, Mobile Developers and Developer Desktop. Most of the respondents were Web Developers (63.3%), followed by 30% Mobile Developers and the rest were Desktop Developer. This reflects the market demand trend is on Web software and Mobile Apps [11][12].

Fig2 explains that the majority of the respondents working for startup companies was programmers Junior (70%). In startup companies, senior programmers are usually the leaders. This phenomenon determines the perspectives for the OOP considered difficult among junior programmers [13].

Fig3 shows that 83% of the respondents were male and only 17% of them were female. It was quite difficult to sample female respondents because female employees usually worked as testers or application support. Besides, men are likely more interested in programming than women [14].

Fig4 presents respondents' age. Respondents of this study were relatively young with an average age of 26 years. At this age, people will find it easier to accept or make changes [15] allowing them to swift their preference over programming paradigm to fit the current trend or demand.

Table 2: Normality Test Using One-Sample Kolmogorov-Smirnov Test

		Section. 2; Programmer preferences to programming paradigm	Section 3; Experience of handling projects	Section 4; Experience of handling obstacles	Section 5; Recommendation
N		30	30	30	30
Normal Parameters[a,b]	Mean	15,97	21,50	10,00	20,37
	Std. Deviation	3,528	3,472	3,591	3,178
Most Extreme Differences	Absolute	,108	,177	,190	,146
	Positive	,108	,157	,190	,105
	Negative	-,089	-,177	-,082	-,146
Test Statistic		,108	,177	,190	,146
Asymp. Sig. (2-tailed)		,200[c,d]	,018[c]	,007[c]	,104[c]

Table 3: Correlation 1

			Programmer preferences to programming paradigm	Experience of handling projects
Spearman's rho	Programmer preferences to programming paradigm	Correlation Coefficient	1,000	,418*
		Sig. (2-tailed)	.	,022
		N	30	30
	Experience of handling projects	Correlation Coefficient	,418*	1,000
		Sig. (2-tailed)	,022	.
		N	30	30

*. Correlation is significant at the 0.05 level (2-tailed).

Table 4: Correlation 2

			Programmer preferences to programming paradigm	Experience of handling obstacles
Spearman's rho	Programmer preferences to programming paradigm	Correlation Coefficient	1,000	-,161
		Sig. (2-tailed)	.	,396
		N	30	30
	Experience of handling obstacles	Correlation Coefficient	-,161	1,000
		Sig. (2-tailed)	,396	.
		N	30	30

*. Correlation is significant at the 0.05 level (2-tailed).

Table 5: Correlation 3

		Programmer preferences to programming paradigm	Recommendation
Programmer preferences to programming paradigm	Pearson Correlation	1	,238
	Sig. (2-tailed)		,206
	N	30	30
Recommendation	Pearson Correlation	,238	1
	Sig. (2-tailed)	,206	
	N	30	30

To see the correlation that occurs, normality test was performed, which results are as follows.

- **Programmers' preferences of programming paradigm**
  asymp.sig.(2-tailed) > α (0,200 > 0,05) means the data of programmer preferences of programming paradigm is normally distributed.
- **Experience of handling projects**
  asymp.sig.(2-tailed) < α (0,018 < 0,05) means the data of experience of handling projects is not normally distributed.
- **Experience of handling obstacles**

asymp.sig.(2-tailed) < α (0,007 < 0,05) means the data of experience of handling obstacle is not normally distributed.

- **Recommendation**
  asymp.sig.(2-tailed) > α (0,104 > 0,05) means the data of recommendation is normally distributed.

3.2 Discussion

After performing the normality test, the correlations between programming paradigm with other variables including experience in handling projects, the experience of dealing with constraints and also to the recommendations were tested.

As shown in Table 2, data on programmers' preference were normally distributed, but the data on the experience in managing projects were not normally distributed. Hence, nonparametric statistical method was administered in the form of Spearman rank correlation.

The Spearman rank method shows these following interpretations.

- If sig. (2-tailed) < α means a significant or influential
- If sig. (2-tailed) > Means α insignificant or less influential
- The level of significance (α) = 0.05

Based on Table 3, the Spearman rank correlation coefficient was (rs) = 0.418 and sig. (2-tailed) = 0,022. sig. (2-tailed) <α (0.022 <0.05), indicating that programmers' preferences affected their experiences in handling projects. Better preference resulted in better competence in handling programming projects.

The positive significant correlation coefficient (0.418) indicates that respondents with strong experience in handling project tend to score programmer preferences against programming paradigms high. Whereas, respondents with low scores in experience of handling projects tend to score programmer preferences for programming paradigms low.

Table 4 shows that data on programmers' preference were normally distributed, yet the data on the experience in handling projects were not. Hence, nonparametric statistical method was used in the form of Spearman rank correlation test. From the table 4 we get the Spearman rank correlation coefficient (rs) = -0161 And sig. (2-tailed) = 0396. Which is sig. (2-tailed)> α (0.396> 0.05) means that a given paradigm preference over problem experiences has little effect. The preference that will get better is not necessarily in line with experience in dealing with problems because the problems faced are not always the same so that the handling of the problems has less effect.

As presented in Table 5, data related to programmers' preference have been normally distributed, and parametric statistical method was used in the form of Pearson correlation analysis. The Pearson correlation coefficient (r) = 0.238 and sig. (2-tailed) = 0.206. sig. (2-tailed)> α (0.206> 0.05) shown in Table 4 indicate the presence of a significant correlation between programmers' preference to programming paradigms and their recommendations.

The correlation coefficient of 0.238 did not show significant positive recommendation that respondents with high scores did not necessarily reflect strong programming paradigm, vice versa.

There was a significant correlation between the preferences of the programmer and the programming paradigm with experience handling projects. More experienced programmers had strong tendency to choose OOP as a preferred paradigm in handling software development projects. The reusability of OOP might be a factor that infleuences programmers' preference over programming paradigms because it can increase productivity and reduce the cost of the project [16].

There is no significant relationship found between programmers' preference to the programming paradigm with constraints. More experienced programmers did not show strong relationship with OOP implementation constraints in their work environment. Senior and junior programmers did not really find difficulties in using OOP. Possible constraints occurred only during the early process of learning OOP when they had to learn about the concept of constructors, encapsulation, and parameters of a class of objects [17].

There is no significant relationship found between programmers' preference to the programming paradigm with the recommendations. It can be implied that greater programming experience was not necessarily correlated with the recommendations. Thus, programmers did not always use OOP as solution in completing programming projects and as the trend in the future.

Programmers might grow interest in other paradigms such as functional programming which also shows a positive trend yet it has not yet been much-discussed among academics or practitioners [18].

4. CONCLUSIONS

This study concluded that OOP paradigm is still the most-preferred paradigm in the IT industry in Yogyakarta and it has been a factor that influences programmers' preferences in Yogyakarta.

Wide knowledge of programming paradigm generates OOP preference as an option. Having experience in handling projects has resulted in a statement that most programmers still rely on OOP in software development at this time, but they also not sure that OOP still will be the first choice for each development software in the future.

This study has limitations on the number of respondents and IT companies involved as samples. It is considered necessary to investigate why respondents had doubt that OOP could be a good choice in working on future projects. In addition, paradigms that are deemed suitable based on practitioners' perspectives are also worth researching.

REFERENCES

[1] M. H. van Emden, "Object-oriented programming as the end of history in programming languages," in 1997 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, PACRIM. 10 Years Networking the Pacific Rim, 1987-1997, 1997, vol. 2, pp. 981–984, doi: 10.1109/PACRIM.1997.620424.

[2] A. Urdhwareshe, "Object-Oriented Programming and its Concepts," *Int. J. Innov. Sci. Res.*, vol. 26, no. 1, pp. 1–6, 2016.

[3] S. Samuel and S. Gorai, "The impact of programming paradigm concepts in mobile application design and its resource utilizations," in *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, 2013, pp. 355–359, doi: 10.1109/ICCSNT.2013.6967128.

[4] O. Camp and R. Woodward, "ICT ENHANCED LEARNING EXPERIENCE FOR AN INTRODUCTORY OBJECT ORIENTED PROGRAMMING COURSE - A Case Study," in *Proceedings of the 3rd International Conference on Computer Supported Education*, 2011, vol. 2, pp. 16–25, doi: 10.5220/0003297000160025.

[5] J. Stevenson and M. Wood, "Recognising object-oriented software design quality: a practitioner-based questionnaire survey," *Softw. Qual. J.*, vol. 26, no. 2, pp. 321–365, Jun. 2018, doi: 10.1007/s11219-017-9364-8.

[6] I. Kaur, N. Kaur, A. Ummat, J. Kaur, and N. Kaur, "Research Paper on Object Oriented Software Engineering," *Int. J. Comput. Sci. Technol.*, vol. 7, no. 4, pp. 36–38, 2016.

[7] H. Jusuf, N. Ibrahim, and A. Atwi Suparman, "Development of Hybrid Learning Material to Support Object-Oritented Programming Course," in *Proceedings of the International Conference of Science and Technology for the Internet of Things*, 2019, doi: 10.4108/eai.19-10-2018.2281285.

[8] L. L. P. TINGGI and W. V YOGYAKARTA, "Statistik PDDIKTI LLDIKTI Wilayah V," *Statistik PDDIKTI*, 2019. [Online]. Available: https://lldikti5.kemdikbud.go.id/home/statpddikti.

[9] M. A. Zaky, I. Nuzar, W. E. Saputro, B. D. S. Praysuta, S. B. Wijaya, and M. Riswan, "Mapping & Database Startup Indonesia 2018," *Badan Ekon. Kreat.*, pp. 1–202, 2018.

[10] KOMINFO, *Daftar Unit Kompetensi Okupasi Dalam Kerangka Kualifikasi Nasional Indonesia Bidang Teknologi Informasi dan Komunikasi (TIK)*, Ver.03-300. Jakarta: KOMINFO, 2018.

[11] M. Jazayeri, "Some Trends in Web Application Development," in *Future of Software Engineering (FOSE '07)*, 2007, no. June 2007, pp. 199–213, doi: 10.1109/FOSE.2007.26.

[12] M. Y. Sharma, M. Bharat, K. Dak, and M. N. Acharya, "Emerging trends in mobile apps market and their potential impact on mobile users engagement in the global economy," *Pune Annu. Res. J. Symbiosis Cent. Manag. Stud. Pune*, vol. 5, no. March, pp. 61–81, 2017.

[13] M. Vujošević-Janičić and D. Tošić, "The role of programming paradigms in the first programming courses," *Teach. Math. XI*, vol. 2, pp. 63–83, 2008.

[14] A. Yadav, S. Gretter, and J. Good, "Computer Science for All: Role of Gender in Middle School Students' Perceptions About Programming," no. April, 2017.

[15] F. Kunze, S. Boehm, and H. Bruch, "Age, resistance to change, and job performance," *J. Manag. Psychol.*, vol. 28, no. 7/8, pp. 741–760, Nov. 2013, doi: 10.1108/JMP-06-2013-0194.

[16] P. Paetau, "Object-Oriented Software Reuse – On The Benefits and Problems," pp. 1–17, 2005.

[17] S. M. Biju, "Difficulties in Understanding Object Oriented Programming Concepts," in *Lecture Notes in Electrical Engineering*, vol. 152 LNEE, no. August 2013, 2013, pp. 319–326.

[18] A. Khanfor and Y. Yang, "An Overview of Practical Impacts of Functional Programming," in *2017 24th Asia-Pacific Software Engineering Conference Workshops (APSECW)*, 2017, vol. 2018-Janua, no. December 2017, pp. 50–54, doi: 10.1109/APSECW.2017.27.