Universal Journal of Educational Research 8(9A): 10-15, 2020 http://www.hrpub.org
DOI: 10.13189/ujer.2020.082002

Programmer's Competencies between Industry and Education

Lucia Sri Istiyowati[1,2,*], Zulfiati Syahrial[1], Suyitno Muslim[1]

[1]Faculty of Educational Technology, Universitas Negeri Jakarta, Indonesia
[2]Faculty of Information Technology, Perbanas Institute, Jakarta, Indonesia

Received December 10, 2019; Revised June 20, 2020; Accepted July 29, 2020

Cite This Paper in the following Citation Styles

(a): *[1] Lucia Sri Istiyowati, Zulfiati Syahrial, Suyitno Muslim , "Programmer's Competencies between Industry And Education," Universal Journal of Educational Research, Vol. 8, No. 9A, pp. 10 - 15, 2020. DOI: 10.13189/ujer.2020.082002.*

(b): *Lucia Sri Istiyowati, Zulfiati Syahrial, Suyitno Muslim (2020). Programmer's Competencies between Industry And Education. Universal Journal of Educational Research, 8(9A), 10 - 15. DOI: 10.13189/ujer.2020.082002.*

Abstract Nowadays the industry of Information Technology (IT) requires a lot of experts on programming but they find it is difficult to hire ones. On the other hand, universities produce many graduates in this field of programming. There is a gap or competency mismatch between computer science graduates/IT graduates with the competencies needed by industry. This phenomenon is a challenge for universities to produce graduates who have abilities that are in accordance with the standards or competencies needed by the industry. The challenge for universities is creating a competent, acceptable and highly competitive workforce by creating learning that can attract students to pursue their competencies. The purpose of this study is to determine the perceptions of lecturers and professionals in the IT field about the competencies that must be possessed by a programmer through questionnaires distributed to industry and universities. The results of the study show that the competencies of programmers according to industry users, professionals and teaching lecturers do not have a big difference, which means that the learning objectives of programming in higher education are in line with what is desired by the industry as a graduate user. In addition to hard skills and soft skills, currently, a programmer also needs to have business skills.

Keywords Competency, Programmer, Industry, Education

1. Introduction

Information Technology affected many areas and human life caused closeness and dependency of humans on technology is increasingly high. This has caused the need for programmers to increase, while Indonesia's human resources in the field of information technology have not been able to meet the needs of industries (Agung, 2017)(Triwijanarko, 2019). The need for this workforce must be balanced with the creation of competent workers. Currently, the world of work, especially in the IT field requires a lot of experts in the field of programming but it is still difficult to get it, on the other hand, many universities produce graduates in that field. This is due to the lack of standards for the ability of graduates in computer science / IT in Indonesia. Many students choose not to wrestle with coding when they are in college so that itraises the phenomenon that very few IT graduates can program when graduating (Senjaya, 2015).

This phenomenon is important for universities, especially in Indonesia, to produce graduates who have the ability that is in accordance with the standards or competent in their fields. It is a big challenge for universities in preparing their graduates to be absorbed and accepted by the appropriate industry. Higher education is challenged to create a competent workforce that is acceptable and highly competitive by creating a learning community that can support students to become graduates with abilities that can be accepted by prospective users.

Education in the field of computer science must be able

to prepare a student to be a more holistic workforce not just those with the technical ability (ACM Computing Curricula Task Force, 2013, p. 15) Developments and changes in technology that continue to occur must be anticipated and faced properly by students who will later become professionals. For this reason, a solid foundation is needed for them to face the ongoing changes. One of the challenges in computer education is programming, the challenge of understanding the process of making programs and the practice of a programmer to transfer knowledge and skills effectively. Programming activities starts from understanding the main problem in building a systematic method to solve the problem automatically using a particular programming language. In addition to competency programming, it is also an educational challenge in this field along with the changes that always occur a person is required to be always up to date (McGettrick et al., 2005, pp. 45–46). Programming courses are core courses in the informatics study program.

Learning programming or "coding" is known to be difficult and challenging (Tan, Ting, & Ling, 2009, pp. 42–46) (Vujošević-Janičić & Tošić, 2008) (Robins, 2015) (Le, 2016) (Xinogalos, 2014), even for students majoring in informatics. This assumption makes students become "antipathy" and less / dislikes programming courses offered by study programs, while the need for programmers is increasing. Learning programming is difficult because students must learn both cognitive and practical activities, which are foreign elements to beginners (Nuutila, Törmä, Kinnunen, & Malmi, 2008, pp. 47–67).

Research conducted by J. Bennedsen and ME Caspersen states that the graduation rate for programming courses is 69% (12), which was then re-examined by Watson and Li (2014) with results that are not much different, the graduation rate of this course at almost all countries in the world are almost the same, around 67.7% (13). Research on programming learning has been done a lot, but there is still no agreed reason why many novice programmers fail to learn program, what educators must do, or even how topics are arranged in a curriculum (8).

To meet the needs of the industry for a suitable workforce, a common perception is needed between the industry as a user and universities as a workforce provider. At present many industries require programmers, on the other hand education especially in the field of IT provides many graduates but still between needs and providers is not well connected. Competency-based learning aims at achieving student competency at the end of learning. All components are determined based on their suitability for competency achievement. Goal setting is important in learning design. So that learning objectives can be properly identified, it requires participation from various parties including the users of graduates and educators.

2. Materials and Methods

2.1. Programmer Competencies

The notion of competence was stated a lot either by experts, working groups or government which states that competence is knowledge, skills, and attitudes (behaviors) that are important in a professional context in accordance with standards set((Kaur & Gaur, 2017)(Gosselin, 2013) (Nurwardani et al., 2016). The programmer's level of ability varies, Wilson divides programmers into 5 levels namely Novice, Advanced Beginner, Competence, Proficiency, and Expert. Most graduates of Higher Education are expected to reach a competent level. Competent level means being able to choose an organized plan to achieve goals after consideration of the whole situation (Winslow, 1996). Indonesian competency map classifies programmers into different levels and groups of competencies. The competency map is an official document compiled collectively by stakeholders as a reference in meeting various needs related to human resource development, especially in the field of ICT in Indonesia (*Peta Okupasi dalam Kerangka Kualifikasi Nasional Indonesia Bidang Teknologi Informasi dan Komunikasi*, 2017). Competencies that must be possessed by a programmer by Ekpereka (2017) are grouped into 3 parts, namely hard skill, soft skill and business skill (Ekpereka & Chibuogwu, 2017).

2.2. Method

The objective of this study was to compare the perspectives of stakeholders and educators about the competencies that must be possessed by a programmer. This research was conducted with a quantitative approach, which is the process of finding knowledge using data in the form of numbers as a means of analyzing information about what you want to know. A literature study to obtain the competencies that must be possessed by a programmer, these competencies are used as instruments that are then disseminated to respondents.

The population of this research is those who have an interest in the competence of a programmer, namely lecturers, practitioners in the IT field and/or those involved in both jobs.

2.2.1. Instrument

The instrument used was built from existing studies and Indonesian national occupational maps for the field of ICT. Instrument testing is carried out by expert testing. The data of this study are primary data, obtained from the results of questionnaire distribution. The data collected is processed and analyzed descriptively.

2.2.2. Data Collection

For collecting data, the survey was sent to respondent via form.google.com. The respondent was distributed to 3 (three) stakeholder groups, namely:

- lecturer groups are those who only teach on campus
- groups of lecturers and professionals/practitioners are those who teach on campus and also work as professionals in the IT field.
- Industrial groups are professionals/practitioners involved in the field of work related to IT, especially programming

3. Result and Discussion

The participants of this study as shown in figure 1 were 58 which were divided into 3 (three) groups of participants, i.e. lecturer (14 respondents), professional in the industry epecially in IT field (26 respondents) and the group those involved in both professions (26 respondents).

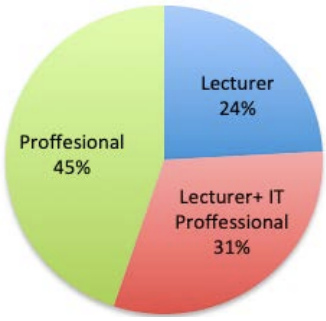

Figure 1. Graph of respondent composition

The results of the study are shown in Figure 2. It can be seen in the graph that in general, the level of agreement between the four groups of respondents regarding these competencies did not have a significant difference. This means that academics and the industrial world have similar point of view about the competencies that must be possessed by a programmer. Another thing that can be drawn from the data obtained is that lecturers who are also involved in IT work have their approval level closer to what is expected by the industrial world.

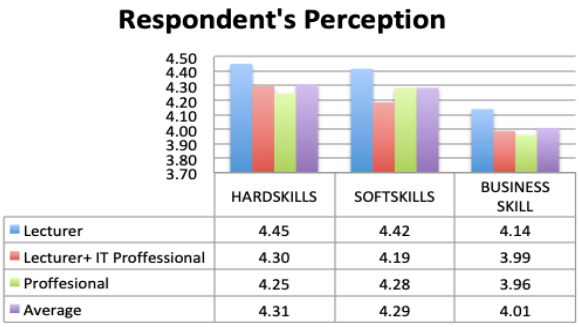

Figure 2. Graph of respondent's perception

In the Hard skills group shown in table 1, competencies whose approval is less than 4 (agree) are competencies H32, H35 and H39.

Table 1. Hard skill competency approval level

CODE	STAKEHOLDERS			AVG
	1	2	3	
H01	4.64	4.39	4.65	4.57
H02	4.71	4.22	4.08	4.28
H03	4.50	4.28	4.42	4.40
H04	4.29	4.28	4.31	4.29
H05	4.79	4.67	4.58	4.66
H06	4.71	4.67	4.23	4.48
H07	4.14	4.61	3.88	4.17
H08	4.50	4.17	4.08	4.21
H09	4.43	4.00	3.96	4.09
H10	4.43	4.06	4.62	4.40
H11	4.43	4.28	4.08	4.22
H12	4.50	4.61	4.23	4.41
H13	4.79	4.67	4.62	4.67
H14	4.57	4.44	4.38	4.45
H15	4.64	4.56	4.35	4.48
H16	4.57	4.39	4.31	4.40
H17	4.79	4.50	4.27	4.47
H18	4.50	4.44	4.46	4.47
H19	4.43	4.44	4.38	4.41
H20	4.43	4.11	4.00	4.14
H21	4.57	4.44	4.54	4.52
H22	4.29	4.22	4.50	4.36
H23	4.43	4.50	4.38	4.43
H24	4.71	4.61	4.54	4.60
H25	4.71	4.72	4.58	4.66
H26	4.57	4.44	4.38	4.45
H27	4.71	4.56	4.50	4.57
H28	4.21	4.00	4.15	4.12
H29	4.29	4.00	4.23	4.17
H30	4.21	4.06	4.12	4.12
H31	4.50	4.22	4.12	4.24
H32	4.14	4.06	3.73	3.93
H33	4.07	4.33	4.04	4.14
H34	4.29	4.22	4.08	4.17
H35	3.93	3.39	3.81	3.71
H36	4.43	4.00	3.96	4.09
H37	4.21	3.94	3.92	4.00
H38	4.43	4.50	4.42	4.45
H39	4.07	3.56	3.73	3.76
	4.45	4.30	4.25	4.31

(1)Lecturer; (2)Lecturer+Professional; (3)Industry

In the Soft skill group as shown in table 2, competencies

whose approval is less than 4 (agree) are competencies S40, S41 and S51.

Table 2. Soft skill competency approval level

CODE	STAKEHOLDERS			AVG
	1	2	3	
S40	4.07	3.89	3.88	3.93
S41	4.07	3.83	3.92	3.93
S42	4.29	3.94	4.15	4.12
S43	4.57	4.50	4.31	4.43
S44	4.43	4.61	4.42	4.48
S45	4.57	4.11	4.42	4.36
S46	4.21	4.06	4.27	4.19
S47	4.36	4.17	4.27	4.26
S48	4.36	4.11	4.27	4.25
S49	4.21	3.78	4.08	4.02
S50	4.64	4.50	4.46	4.52
S51	4.14	3.67	3.88	3.88
S52	4.50	4.39	4.54	4.48
S53	4.64	4.56	4.58	4.59
S54	4.64	4.17	4.46	4.41
S55	4.57	4.22	4.46	4.41
S56	4.71	4.50	4.69	4.64
S57	4.50	4.33	4.04	4.24
	4.42	4.19	4.28	4.29

(1)Lecturer; (2)Lecturer+Professional; (3)Industry

In the Business skills group as shown in table 3, competencies whose approval is less than 4 (agree) are competencies B58, B60, B61, B64, B67, B68 AND B73.

Table 3. Business skill competency approval level

CODE	STAKEHOLDERS			AVG
	1	2	3	
B58	4.00	3.89	3.81	3.88
B59	4.50	4.50	4.54	4.52
B60	4.36	3.67	3.54	3.78
B61	4.21	4.06	3.73	3.95
B62	4.21	3.89	4.15	4.09
B63	4.29	4.17	4.08	4.16
B64	3.71	3.06	3.19	3.28
B65	3.86	4.00	4.08	4.00
B66	4.00	4.17	4.00	4.05
B67	3.86	3.83	3.88	3.86
B68	4.00	3.94	3.69	3.84
B69	4.29	3.94	3.96	4.03
B70	4.57	4.65	4.42	4.53
B71	4.36	4.22	4.12	4.21
B72	4.00	4.00	4.08	4.03
B73	4.00	3.83	4.08	3.98
	4.14	3.99	3.96	4.01

(1)Lecturer; (2)Lecturer+Professional; (3)Industry

From the results of processing the questionnaire data, competencies were taken which were on agreed by all groups / respondents. The Likert scale stating that a value of 4 means agree, based on that the competencies required by the programmer are taken who have an average value of 4 (Agree). Declaration of competency less than 4 is eliminated from the level of competence that must be posed by a programmer. Programmers' competencies that were approved by respondents as shown in table 4.

Table 4. List of programmer competencies

NO.	CODE	Avg. Level of agreement	COMPETENCY
	H01	4.57	Analyzing problems
	H02	4.28	Analyzing users' software requirements
	H03	4.40	Analyzing work processes, develop work processes and discover new approaches to solve them efficiently within the deadlines
	H04	4.29	Build various concepts, procedures, models and diagrams (such as flowcharts) that instruct how to write software code
	H05	4.66	Use existing algorithms and code to solve similar problems (reuse)
	H06	4.48	Use the right tools (software that are used to execute source code programming)
	H07	4.17	Cluster programming tools algorithmically into a series
	H08	4.21	Explore and evaluate application design considerations for various technologies
	H09	4.09	Use the latest application design methodologies, tools and techniques to transform business requirements and logical models into technical application design
	H10	4.40	Search for qualified information as required to develop software quickly
	H11	4.22	Recognizing computer capabilities in order to design effective software
	H12	4.41	Write scripts (code) that can automate solutions by using a basic scripting process and options on application-specific scripting
	H13	4.67	Use Data Structures

Table 4 continued

	H14	4.45	Use SQL
	H15	4.48	Implement database access
	H16	4.40	Use libraries or pre-existing components
	H17	4.47	Develop data-based applications
	H18	4.47	Develop incrementally (Incremental development)
	H19	4.41	Create Model-based programming (Model-based programming)
	H20	4.14	Build user interfaces (user interfaces / UI)
	H21	4.52	Repair and modify algorithm / code that is not good enough to fit the problem being solved
	H22	4.36	Refactoring (The process of clarifying and simplifying the design of existing code, without changing its behavior)
	H23	4.43	Test program quickly and efficiently
	H24	4.60	Debug program quickly and efficiently
	H25	4.66	Handle errors (error handling)
	H26	4.45	Implement structured programming
	H27	4.57	Implement object-oriented programming
	H28	4.12	Document every aspect of an application or system as a reference for future maintenance and improvement
	H29	4.17	Provide support for implementation and applications problem solving
	H30	4.12	Monitor and improve application performance
	H31	4.24	Use Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS) as insight into web design and development
	H33	4.14	Use Java Script
	H34	4.17	Use server language, to develop a complete chain from front-end applications through the API layer and to the back-end
	H36	4.09	Implement software development cycles
	H37	4.00	Measure the quality of software needed for real-time applications
	H38	4.45	Modify programs efficiently
	S42	4.12	Adapt to the organization's culture and policies
	S43	4.43	Confident with personal ideas but stay open for feedback
	S44	4.48	Adapt to changes quickly and stay focused on project priorities (meeting deadlines)
	S45	4.36	Create and to innovate / to initiate "out of the box" ideas and apply them to achieve results
	S46	4.19	Good memory skills
	S47	4.26	Express ideas
	S48	4.25	Maintain strong relationships with various parties in the organization
	S49	4.02	Provide mentoring and support to clients
	S50	4.52	Think critically
	S52	4.48	Work within a high level of accuracy and a systematic approach for continuous checking and testing
	S53	4.59	Concentrate and attention to details
	S54	4.41	Identify and articulate problems
	S55	4.41	Solve problems and encounter barriers at work effectively
	S56	4.64	Work with others (teamwork)
	S57	4.24	Actualize abstract ideas
	B59	4.52	Collaborate with other programmers during project development
	B62	4.09	Empathize with end users, understanding the business conditions in work
	B63	4.16	Explain to their customers how the software works and answer any questions that arise
	B65	4.00	Use foreign languages
	B66	4.05	Recommend "upgrades" of existing software programs and systems
	B69	4.03	Manage crises arouses from team arguments and move forward to complete the task at hand
	B70	4.53	Work in accordance with software/user security regulations
	B71	4.21	Identify and correct gaps in a project
	B72	4.03	Do predictive modeling (models that show what will happen if something happens)

4. Conclusion

Based on the results of studies that have been carried out, it can be seen that the competencies of programmers according to industry users, professionals and teaching lecturers do not have a big difference, which means that the learning objectives of programming in higher education are in line with what is desired by the industry as a graduate user. In addition to hard skills and soft skills, currently, a programmer also needs to have business skills. The limitation of this study is that there has been no reconfirmation with the industry whether the competency used as an instrument is sufficient as the competency of a programmer. For further research, a deeper discussion with the industry is needed about whether the competencies that have been generated from this research are sufficient to meet the criteria of a programmer. It is also necessary to examine the learning process that occurs to evaluate the results of the achievement of learning objectives so that graduates can be accepted by the industry.

REFERENCES

[1] ACM Computing Curricula Task Force (Ed.). (2013). *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. Retrieved from http://dl.acm.org/citation.cfm?id=2534860

[2] Agung, B. (2017, July 28). Indonesia Darurat Tenaga Programmer. Retrieved April 10, 2019, from teknologi website: https://www.cnnindonesia.com/teknologi/20170728094848-185-230919/indonesia-darurat-tenaga-programmer

[3] Ekpereka, I., Nnenna, & Chibuogwu, C. I. (2017). Computer Programming Competencies Required by Computer Education Graduates for Sustainable Employment. *Review of European Studies*, *9*(2), 106. https://doi.org/10.5539/res.v9n2p106

[4] Gosselin, D. (2013, June). Competencies vs Learning Outcomes. Retrieved December 4, 2018, from Strengthen Workforce Prep in your Program website: https://serc.carleton.edu/integrate/programs/workforceprep/competencies_and_LO.html

[5] Kaur, R., & Gaur, R. (2017). ICT Skills and Competencies for LIS Professionals in Digital Environment. *International Journals of Research in Engineering, IT and Social Science*, *7*(6), 92–95.

[6] Le, N.-T. (2016). A Classification of Adaptive Feedback in Educational Systems for Programming. *Systems*, *4*(2), 22. https://doi.org/10.3390/systems4020022

[7] McGettrick, A., Boyle, R., Ibbett, R., Lloyd, J., Lovegrove, G., & Mander, K. (2005). Grand Challenges in Computing: Education—A Summary. *The Computer Journal*, *48*(1). https://doi.org/10.1093/comjnl/bxh064

[8] Nurwardani, P., Ridwan Tutupoho Roy, Mulyono, E., Wahyu, S., Endrotomo, & Arifin. (2016). *Buku Panduan Penyusunan Kurikulum Pendidikan Tinggi* (Kedua). Jakarta: RISTEKDIKTI.

[9] Nuutila, E., Törmä, S., Kinnunen, P., & Malmi, L. (2008). Learning Programming with the PBL Method — Experiences on PBL Cases and Tutoring. In *Lecture Notes in Computer Science*. *Reflections on the Teaching of Programming*. Retrieved from https://link.springer.com/chapter/10.1007/978-3-540-77934-6_5

[10] *Peta Okupasi dalam Kerangka Kualifikasi Nasional Indonesia Bidang Teknologi Informasi dan Komunikasi*. (2017). Kominfo.

[11] Robins, A. (2015). The ongoing challenges of computer science education research. *Computer Science Education*, *25*(2), 115–119. https://doi.org/10.1080/08993408.2015.1034350

[12] Senjaya, A. (2015, April 3). Programmer dan Pertumbuhan Startup Teknologi Indonesia. Retrieved July 4, 2017, from LinkedIn Pulse website: https://www.linkedin.com/pulse/programmer-dan-pertumbuhan-startup-teknologi-andreas-senjaya

[13] Tan, P. H., Ting, C. Y., & Ling, S. W. (2009). Learning Difficulties in Programming Courses: Undergraduates' Perspective and Perception. *2009 International Conference on Computer Technology and Development*, *1*. https://doi.org/10.1109/ICCTD.2009.188

[14] Triwijanarko, R. (2019, November 1). Indonesia Masih Butuh Banyak Tenaga Programmer. Retrieved November 4, 2019, from Marketeers - Majalah Bisnis & Marketing Online - Marketeers.com website: https://marketeers.com/indonesia-masih-butuh-banyak-tenaga-programmer/

[15] Vujošević-Janičić, M., & Tošić, D. (2008). The role of programming paradigms in the first programming courses. *The Teaching of Mathematics*, (21), 63–83.

[16] Winslow, L. E. (1996). Programming pedagogy—a psychological overview. *ACM Sigcse Bulletin*, *28*(3), 17–22.

[17] Xinogalos, S. (2014). Designing and deploying programming courses: Strategies, tools, difficulties and pedagogy. *Education and Information Technologies*, *21*(3), 559–588. https://doi.org/10.1007/s10639-014-9341-9