



Faculty of Computer Science and Information Technology

**EVALUATING THE IMPACT OF USING GITHUB
COPILOT BY COMPUTER SCIENCE STUDENTS IN
UNIVERSITI MALAYSIA SARAWAK**

Etheldritha Lim Han Ching

Bachelor of Computer Science with Honors
(Computational Science)

2023

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

EVALUATING THE IMPACT OF USING GITHUB COPILOT AMONG COMPUTER SCIENCE STUDENTS IN UNIVERSITI MALAYSIA SARAWAK

ACADEMIC SESSION: 2022/23

ETHELDRITHA LIM HAN CHING (CAPITAL LETTERS)

hereby agree that this Thesis* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

- 1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [or for the purpose of interlibrary loan between HLI]
5. ** Please tick (v)

Form with checkboxes for CONFIDENTIAL, RESTRICTED, and UNRESTRICTED, each with a brief description of the classification level.

Handwritten signature of the author.

(AUTHOR'S SIGNATURE)

Permanent Address

NO. 1, SAMPLE PARK, JALAN TUN HUSSIEN ONN, 97000, BINTULU, SARAWAK

Validated by

Handwritten signature of the supervisor.

Assoc. Prof. Dr. Johari Abdullah Faculty of Computer Science & Information Technology Universiti Malaysia Sarawak SUPERVISOR'S SIGNATURE

Date: 24/7/2023

Date: 24/7/2023

Note * Thesis refers to PhD, Master, and Bachelor Degree ** For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

**EVALUATING THE IMPACT OF USING GITHUB
COPILOT BY COMPUTER SCIENCE STUDENTS IN
UNIVERSITI MALAYSIA SARAWAK**

ETHELDRITHA LIM HAN CHING

This project is submitted in partial fulfillment of the
requirements for the degree of
Bachelor of Computer Science with Honors
(Computational Science)

Faculty of Computer Science and Information
Technology

UNIVERSITI MALAYSIA SARAWAK

2023

**MENILAI IMPAK PENGGUNAAN GITHUB COPILOT KEPADA
PELAJAR-PELAJAR SAINS KOMPUTER DI UNIVERISTI
MALAYSIA SARAWAK**

ETHELDRITHA LIM HAN CHING

Projek ini merupakan salah satu keperluan untuk
Ijazah Sarjana Muda Sains Komputer Dengan Kepujian
(Sains Komputan)

Fakulti Sains Komputer dan Teknologi Maklumat
UNIVERSITI MALAYSIA SARAWAK

2023

Declaration

I hereby declare that the thesis entitled 'Evaluating the Impact of Using GitHub Copilot by Computer Science Students in Universiti Malaysia Sarawak' is based on my original work except for quotation and citation which have been duly acknowledging. I also declare that no portion of the work referred to in this report has been submitted in support of an application for another degree at Universiti Malaysia Sarawak (UNIMAS) or qualification of this or any other university or institution of higher learning.



Etheldritha Lim Han Ching (72215)

1 July 2023

Computational Science

Faculty of Computer Science and Information Technology

Universiti Malaysia Sarawak

Acknowledgement

Firstly, I would like to thank God for the patience and determination that he has provided me with and with it I have manage to successfully complete the first part of my Final Year Project. I would like to thank my parents and siblings for their endless encouragement, emotional and financial support for me to complete my tasks. Not forgetting, their constant prayers for my success has guided me to finish and complete this project.

Then, I would like to express my sincere gratitude to my supervisor, Dr. Johari bin Abdullah, for taking the time to discuss the project with me, for guiding me as I conducted this research, and for his extraordinary efforts to provide direction and inspiration in order to make this Final Year Project a complete success. This project proceeded easily and smoothly thanks to his guidance and skills.

Not to forget my fellow roommates and friends who pushed me through finishing this Final Year Project and presented me with all the help I needed. I am forever thankful for all they have done for me.

Abstract

Programming or coding is an important skill that has to be learned by computer science students. There has been a couple of coding assistance tools in the market. The latest one is Artificial Intelligent driven code completion tools, such as GitHub Copilot that help students write code with just a simple natural language command in the form of comment. This tool is a great help for programmers as it fastens the time taken to write a program. However, it is unknown whether this tool can be a great learning tool for students that are still learning and mastering programming skill and languages. This research is done to evaluate the impact and acceptance of using GitHub Copilot among computer science students in Universiti Malaysia Sarawak and to find out the parameters that can be used to evaluate students' efficiency in coding a program. To do this research, a controlled experiment involving a selective group of students who used both GitHub Copilot and conventional programming methods to complete a programming task and then questionnaires will be distributed to the group of students. The questionnaires contain question to find out the students' opinion of the tools and how well they are accepting it using the Unified Theory of Acceptance and Use of Technology (UTAUT). After the students answer the survey, the result will be discussed and analysed. Overall, the result of the evaluation reveals that GitHub Copilot significantly improved the productivity of computer science students. The tool shortens the code-writing process, reducing development time and minimizing the likelihood of syntax errors. Additionally, GitHub Copilot enhanced code quality by suggesting relevant and accurate code snippets. However, some students consider the usage of the tool as cheating and plagiarising. The user satisfaction may vary depending on individual preferences and experiences. Therefore, careful monitoring and guidance are necessary to ensure that students strike a balance between utilizing the tool and developing their own coding expertise.

Keywords: autocomplete tools, GitHub Copilot, impact on students, programming, UTAUT

Abstrak

'Programming' atau 'coding' adalah sebuah kemahiran yang penting untuk dipelajari oleh pelajar komputer sains. Terdapat beberapa alat bantuan 'coding' yang terdapat di pasaran. Sebagai contoh, GitHub Copilot yang membantu pelajar menulis kod mereka dengan hanya menulis komen menggunakan bahasa semula jadi. Alatan ini membantu 'programmer' meningkatkan masa untuk menulis sebuah program. Walau bagaimanapun, sukar untuk mengetahui sama ada alatan ini dapat memberi impact yang positif kepada para pelajar yang masih mempelajari bahasa 'programming'. Project ini dilakukan untuk menilai impact dan penerimaan penggunaan GitHub Copilot di kalangan pelajar-pelajar komputer science di Universiti Malaysia Sarawak dan untuk mengetahui pembolehubah untuk menilai kecekapan pelajar dalam kemahiran 'coding'. Dalam penyelidikan ini, satu eksperimen terkawal melibatkan kumpulan pelajar yang dipilih menggunakan GitHub Copilot dan kaedah pengaturcaraan konvensional untuk menyelesaikan tugas pengaturcaraan, dan kemudian soal selidik akan diedarkan kepada kumpulan pelajar tersebut. Tinjauan tersebut mengandungi soalan untuk mengetahui tentang pendapat pelajar-pelajar terhadap GitHub Copilot and bagaimana penerimaan mereka terhadap alatan tersebut. Selepas itu, keputusan akan dianalisis untuk mengetahui impact dan penerimaan di kalangan pelajar. Secara keseluruhannya, hasil penilaian menunjukkan bahawa GitHub Copilot secara signifikan meningkatkan produktiviti pelajar. Alat ini memendekkan proses menulis 'code' dan mengurangkan kemungkinan kesalahan sintaks. Walau bagaimanapun, sesetengah pelajar menganggap penggunaan alat ini sebagai penipuan dan plagiat. Oleh itu, pemantauan yang teliti diperlukan untuk menangani cabaran dan memastikan pelajar mencapai keseimbangan antara menggunakan alat ini dan mengembangkan kepakaran pengaturcaraan mereka sendiri.

Kata kunci: alat autolengkap, GitHub Copilot, impact, programming, UTAUT

Table of Contents

Declaration	i
Acknowledgement	ii
Abstract	iii
List of Tables	x
List of Figures	xii
List of ABBREVIATIONS	xiv
Chapter 1 Introduction	1
1.1 Introduction	1
1.2 Problem Statement	7
1.2.1 Research Questions	8
1.3 Objectives.....	9
1.4 Research Scope	9
1.5 Project Timeline	10
1.6 Brief Methodology	10
1.7 Expected Outcome	11
Chapter 2 Literature Review	12
2.1 Introduction	12
2.2 Background of Code Completion Tool	13

2.3	Code completion tools and review of existing technology & comparison	15
2.3.1	AI-driven code completion.....	15
2.3.2	Standard code completion	19
2.3.3	Comparison on discussed code completion tools.....	22
2.4	Advantages of using code completion tools.....	23
2.5	Disadvantages of using code completion tools	24
2.6	Related works	26
2.7	Acceptance Model.....	28
2.7.1	Technology Acceptance Model (TAM)	28
2.7.2	Unified Theory of Acceptance and Use of Technology (UTAUT)	29
2.7.3	TAM vs UTAUT	31
2.8	Conclusion.....	32
Chapter 3 Methodology		33
3.1	Introduction	33
3.2	Methodology flowchart	34
3.3	Sampling Method	36
3.3.1	Targeted respondent	36
3.4	Experiment: Coding using GitHub Copilot vs without GitHub Copilot.....	36
3.5	Constructing survey questions	37
3.5.1	Respondents' background	37
3.5.2	Respondents' programming skill and their opinion on programming	38

3.5.3	Respondents' opinion on coding with and without GitHub Copilot.....	38
3.5.4	Respondents' opinion on GitHub Copilot.....	38
3.5.5	Respondents' acceptance towards GitHub Copilot.....	39
3.6	Data collection method.....	41
3.6.1	Pilot Test and Reliability Test.....	41
3.7	Research instrument	42
3.8	Data analysis	43
3.8.1	Descriptive analysis.....	43
3.8.2	Exploratory analysis.....	43
3.9	Conclusion.....	44
Chapter 4 Data Analysis and Discussion.....		45
4.1	Introduction	45
4.2	Response Rate	45
4.3	Reliability Analysis	46
4.3.1	Pilot Test	46
4.4	Coding using GitHub Copilot vs without GitHub Copilot analysis.....	47
4.5	Demographic Analysis	48
4.5.1	Gender Analysis	48
4.5.2	Age Analysis	49
4.5.3	Educational Level Before UNIMAS Analysis.....	50
4.5.4	MUET Result Analysis	51

4.6	Summary of Demographic Background.....	52
4.7	Students’ programming skill and their opinion on programming	52
4.8	Students’ opinion on coding with and without GitHub Copilot.....	54
4.9	Students’ opinion on GitHub Copilot.....	55
4.10	Students’ acceptance towards GitHub Copilot analysis using Unified Theory of Acceptance and Use of Technology (UTAUT) model.....	56
4.11	Normality test.....	59
4.12	Relationship between the variables in UTAUT	60
4.13	Conclusion.....	62
Chapter 5 Conclusion and Recommendation		63
5.1	Introduction	63
5.2	Brief discussion and result	63
5.3	Research Objective 1 - To identify the parameters that can be associated to evaluate the impact of using Artificial Intelligence (AI) enabled code completion tool such as GitHub Copilot	65
5.4	Research Objective 2 – To evaluate the acceptance of GitHub Copilot tool for students.....	67
5.5	Research Objective 3 – To evaluate the impact of using GitHub Copilot on students’ coding ability for assignment completion.....	69
5.6	Limitation of study	71
5.7	Recommendation.....	71

5.7.1	Design of the website	73
5.7.2	Targeted users	82
5.7.3	Website link.....	82
5.8	Conclusion.....	82
5.9	Overall conclusion.....	83
References		84
Appendices		89

List of Tables

Table 2.1: Comparison of discussed code completion tools	22
Table 2.2: Works related to the study of code completion tools.....	27
Table 3.1: Justification for questions asked in respondent's background.....	38
Table 4.1: Summary of responds rate.....	46
Table 4.2: Result of Cronbach Alpha test	46
Table 4.3: Summary of average time taken, and number of errors obtained after completing Task A.	47
Table 4.4: Summary of average time taken and number of errors obtained after completing Task B.	47
Table 4.5: Summary of the gender analysis.	48
Table 4.6: Summary of the age analysis.	49
Table 4.7: Summary of educational level before UNIMAS analysis.....	50
Table 4.8: Summary for MUET analysis	51
Table 4.9: Summary of the demographic background analysis.	52
Table 4.10: Students' programming skill and their opinion on programming result.	53
Table 4.11: Students' opinion on coding with and without GitHub Copilot analysis.	54
Table 4.12: Students' opinion on GitHub Copilot analysis	56
Table 4.13: Students' acceptance towards GitHub Copilot analysis using Unified Theory of Acceptance and Use of Technology (UTAUT) model analysis.....	57
Table 4.14: Summary of the mean scores and standard deviations for six variables related to the acceptance of GitHub Copilot.....	58
Table 4.15: Shapiro-Wilk test result.	59
Table 4.16: Correlation test result.	61

Table 5.1: Summary of the result for three objectives 65

List of Figures

Figure 1.1: List of Popular Programming Languages (Stake Overflow, 2021)	3
Figure 2.1: Common architecture of code completion system (Luo, 2017).	14
Figure 2.2: Examples of GitHub Copilot’s generated code by reading user’s comment. (GitHub, 2022).....	16
Figure 2.3: Example of GitHub Copilot auto filling repetitive code. (GitHub, 2022).....	16
Figure 2.4: Example of Amazon CodeWhisperer generates code to use AWS APIs to upload files to Amazon Simple Storage Service (Amazon S3). (Amazon, 2022)	17
Figure 2.5: Example of CodeWhisperer’s security scans to detect vulnerabilities (Amazon, 2022).....	18
Figure 2.6: Example of Tabnine generating code by reading user’s comment (Tabnine, n.d.)	19
Figure 2.7: Example of Tabnine’s code autocomplete function (Tabnine, n.d.).....	19
Figure 2.8: Example of how IntelliCode provides smart contextual suggestions as developer type their code (Microsoft, 2022).....	20
Figure 2.9: Example of IntelliCode’s AI-Assisted Refactoring support (Smacchia, 2021).....	21
Figure 2.10: Example of writing a TensorFlow network in Python and aiXcoder showing the code option (AIXcoder, 2022).	22
Figure 2.11: First modified version of TAM.....	29
Figure 2.12: Final version of TAM	29
Figure 2.13: UTAUT (Marikyan. & Papagiannidis., 2021).....	31
Figure 3.1: The research’s methodology flowchart	35
Figure 3.2: Rule of thumb for Cronbach’s alpha interpretation (Cronbach, 1951).....	42
Figure 3.3: Comparison of statistic software (Masic et al., 2019)	43
Figure 4.1: Result of the Cronbach Alpha test.	46

Figure 4.2: Summary of the gender analysis.....	48
Figure 4.3: Summary of the age analysis.	49
Figure 4.4: Summary of educational level before UNIMAS analysis	50
Figure 4.5: Summary for MUET analysis	51
Figure 5.1: Landing page.	73
Figure 5.2: Getting started page.	74
Figure 5.3: Registration page.	74
Figure 5.4: Login page.	75
Figure 5.5: Community page.....	76
Figure 5.6: Community post page.....	76
Figure 5.7: Website dashboard.....	77
Figure 5.8: User’s profile page.....	78
Figure 5.9: Administrative AI tool list page.	78
Figure 5.10: Administrative save new AI tool detail page.....	79
Figure 5.11: Administrative edit existing AI tool detail page.....	79
Figure 5.12: Administrative community post list.....	80
Figure 5.13: Administrative report list.....	81
Figure 5.14: Administrative report detail page.	81

List of ABBREVIATIONS

UNIMAS	Universiti Malaysia Sarawak
VR	Virtual Reality
IoT	Internet of Things
AI	Artificial Intelligence
ML	Machine Learning
IR	Industrial Revolution
IDE	Integrated Development Environment
AWS	Amazon Web Server
API	Application Programming Interface
VSCoDe	Visual Studio Code
S3	Simple Storage Service
PBD/E	Programming by Demonstration/Examples
TAM	Technology Acceptance Model
UTAUT	Unified Theory of Acceptance and Use of Technology
SPSS	Statistical Package for the Social Sciences
SAS	Statistical Analysis System

Chapter 1 Introduction

1.1 Introduction

The Fourth Industrial Revolution is the present developing environment where technologies like virtual reality (VR), Internet of Things (IoT) and Artificial Intelligence (AI) are integrated into people's daily life in an attempt to make their life and work easier. This is also known as Industry 4.0 and the speed of how it is changing the way people live is unable to be ignore. Those who are able to understand and is knowledgeable in it could benefit from it while those that does are left behind. It is especially impactful in the working environment, for example, automated machine powered by AI do helps increase human's productivity and efficiency (Kalliamvakou., 2022) (Bikse, 2022).

However, the implementation of these disruptive technologies in the workforce can also bring some negative impact such as it put some jobs under the threat of decreasing in demand. Jobs like data entry clerks, payroll clerks, accountants and auditors are decreasing in demand while IT based jobs like software and applications developers, data analysts and scientists and AI and machine learning specialists are increasing in demand with internet of things specialists among the newly emerging professions.

Computer program consists of code written by programmer that is executable on a computer to perform certain tasks. Programming is the process of giving machines a set of instructions that describe how a program should be carried out (Wilkins, 2021). To build an effective computer program, programmers will spend their whole career learning variety of programming language and tools. Programmers will start by using a code editor or IDE to write a source code, which is a collection of code written in programming language. The source code will then be compiled by a compiler, in which means it will be converted into

machine language so machines can understand the instruction and execute the program. Example is the C, C++ and Java. There are also other languages that doesn't require a compiler. They are called interpreted programming languages because they will use interpreter to read and execute the source code. Examples of interpreted programming languages are JavaScript, PHP and Python.

In an article for IEEE Computer Society, Hierso (2022) discussed on the history of programming language. The first programming language was developed by Ada Lovelace and Charles Babbage in 1883. They worked together on the Analytical Engine, a primitive mechanical computer. Then, in 1970, Niklaus Wirth developed Pascal and was the main language used by Apple for early software development. In 1972, Dennis Ritchie developed C, the first high level programming language. It was regarded as closer to human language and less machine-like code. In 1987, C++ emerges as the dominant object-oriented language. Python was created in 1991. It emphasized on code readability. In 1995, Java was created. It was designed to have as few implementation dependencies as possible, meaning that compiled Java code can be run on all platform that support Java without recompilation.

Programming can be breakdown into a few simple steps. For example, for C program, the programmer will need to download and install a software into their computer. Then, they will create and write the code before compiling and running the codes. Debugging will only be needed when bugs appear. These steps are generally the same across all programming language.

In 2021, Stack Overflow, a forum-like website where professional and enthusiast programmers can share their knowledge or ask question regarding programming, generated the Stack Overflow Developer Survey internationally among its user (Stack Overflow, 2021).

Figure 1.1 shows the result of the most popular programming languages the users use. As we can see, JavaScript took the top spot, which is not surprising as most web browsers utilize it and it's one of the easiest languages to learn.

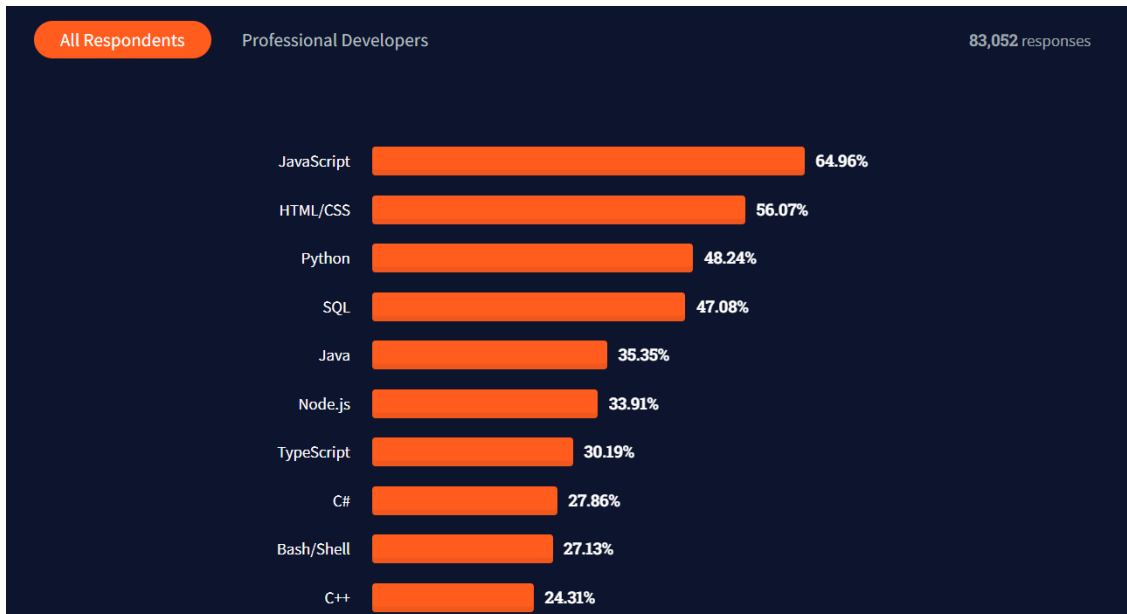


Figure 1.1: List of Popular Programming Languages (Stake Overflow, 2021)

Ministry of Human Resources established the TalentCorp Malaysia's Critical Occupation List (COL) for 2021 (Ministry of Human Resource & Department of Statistic Malaysia, 2022). It is a survey done to find out the occupations in demand within the industry. It aims to create a comprehensive map of Malaysia's most demanded current and future skills and talent towards Industrial Revolution (IR) 4.0. As expected, Computer Science ranked number one on the list which means it was the highest in demand in 2021. The IT industry is growing at a rapid pace. So, universities or higher education institutions need to efficiently produce more computer science graduate and in a quicker pace so they can be in sync with the industry's growth. Thus, educators need to look into ways to equip students with the necessary skill without using up too much time, especially vital skill in computer science like coding a program.

JobStreet is online employment marketplace and is Malaysia’s number 1 job searching platform. Figure 1.2 below is extracted from a page in the JobStreet: Job Outlook Report 2022 (JobStreet, 2022). It shows the Top 10 industries with the most job advertised in the platform. As we can see, Computer or IT industries came in second with 33.6k this year. Based on this statistic, it is believed that the demand for ICT/CS based jobs will only grows in the future and therefore, more computer science graduates will be needed in the market.

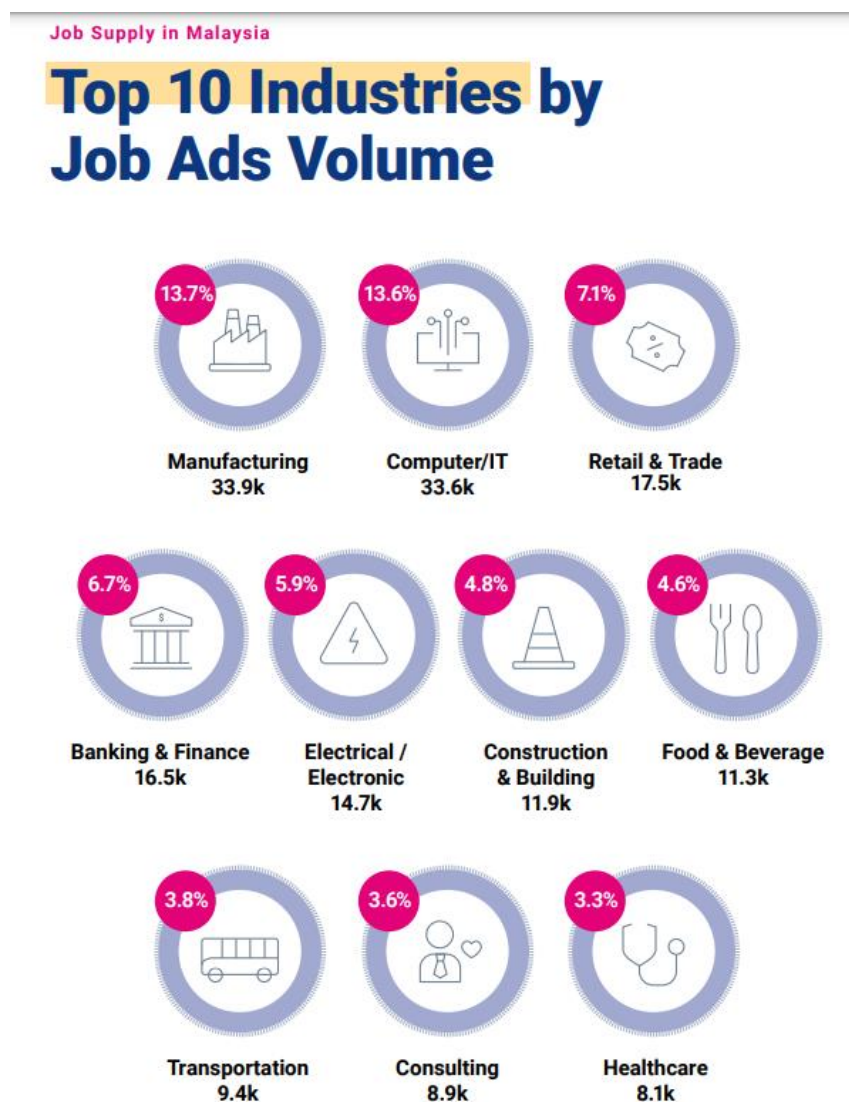


Figure 1.2: List of The Top 10 Industries with Most Job Ads Volume in JobStreet (JobStreet, 2022).

Learning how to code a program using different programming language is an essential part in Universiti Malaysia Sarawak's Computer Science students' study. These students came from different educational background. Some came equipped with basic coding knowledge and some was introduced to it for the first time during their study. This arise the issue of students having a hard time trying to keep up with learning how to code especially those who was not exposed to it properly before.

Students have been taught on how to code by attending classes and have a lab session for an on-hand experience. This way, they are taught the basic and given demonstration on how this code run. Later on, they would be given a programming assignment to test how well they can code. They can also have a self-practice session which could takes a lot of time especially when they encounter bugs or problem. Even so, there are still some that knew how to write a code for a program but is too lazy to type in the line one by one especially when the program requires thousand lines of code.

It's not a surprise that a lot of students struggle with learning how to do programming, especially those who are new to it. Generally, students find programming hard because it'd difficult to memorize the syntax, understanding the structure of program, algorithm construction, understanding the errors and solving the errors that might appear in their program. New learner might also find it hard to understand some programming concepts. For example, parameters, pointers, recursion, data types, repetition, libraries and error handling. Meanwhile, Alhazbi (2016) mentioned that students find programming difficult because "they do not have the problem-solving skills to write logical and correct program algorithms".

Nowadays, with technological advancement, there are several AI pair programming tools developed that could assist programmers especially students in coding their programs. GitHub

Copilot powered by a deep neural network language model called Codex, which was trained on public code repositories on GitHub. This tool offers code generation of lines or even an entire program and also autofill-style suggestions as you code. Programmers receive suggestions from GitHub Copilot either by starting to write the code they want to use, or by writing a natural language in comment form to describe what they want the code to do. It will then analyze the context in the file users are editing, as well as related files, and offers suggestions from within their text editor. GitHub Copilot works well with a lot of programming languages such as Python, JavaScript, TypeScript, Ruby, Go, C#, or C++. This tool is also made available for free for active students all over the world.

GitHub Copilot is an automated code generator that will speed up programming activities for the students. Based on research by Nguyen and Nadi (2022), besides being used for coding, GitHub Copilot can also be used in mathematical problem solving such as linear algebra, probability and even statistical problems.

Therefore, there is no denying that GitHub Copilot could be great tools for students as it could be more efficient especially for those struggling with their code because instead of spending a huge amount of time finding answers on the internet, they can just type in a simple comment in their code and Copilot will display suggested code, as well as completing it for them. In addition, the autocomplete function can help cut short students' time as they wouldn't have to type in the program one by one. However, using an AI-driven tool can also affect the students' productivity and their code's quality. Not only that but this could also affect the students' ability to code out a program.