Basic Statistic Final Report

Decoding Programming Language Trends in Indonesia: Insights from Statistical Analysis

BY:

Ella Raputri (2702298154)

Ellis Raputri (2702298116)

Class L3AC
Computer Science Program
School of Computing and Creative Arts

Bina Nusantara International University
Jakarta
2024

A. BACKGROUND PROBLEM

Nowadays, technology is indispensable in our lives. Various applications and websites have played major roles in our era, enabling the automation of numerous activities requiring significant time and accuracy. From within the infrastructure of those technologies, artificial language exists to describe the underlying process that the computer has to follow. The said artificial language will be translated into machine language and executed by computers [1]. This artificial language is often referred to as the term “programming language”. Programming languages serve as a foundation for a program or software, playing a crucial role in technology growth. They enable developers to create mobile to desktop applications and enforce complex algorithms for data analysis and AI engineering.

For these past decades, the programming language has been evolving rapidly thanks to the involvement of technology growth and innovations. The progression of programming languages can not be separated from the evolution of programming language features throughout the year, where old programming languages acquire new features and new programming languages for specific problems are born. As a result, there exists an abundant amount of programming languages out there, with different features and characteristics. Due to the dynamic and fast-paced nature of the technology industry, programming languages have also experienced major popularity adjustments over time. Some general-purpose programming languages may not address problems as profoundly as the new domain-specific languages [2]. Hence, many of them are eliminated and replaced by new emerging languages.

In Indonesia, the significance of programming languages is also indisputable as in recent years, the Ministry of Industry has devised a roadmap policy to achieve digital transformation in Indonesia 4.0 [3]. In other words, Indonesia has seen potential influences of digital technology development in various sectors, from economy, fintech, and politics, to culture and education. Sectors like fintech, commerce, and education have been the leading sector in instigating demands for skilled programmers. The government initiatives of “Making Indonesia 4.0”, accompanied by the digital growth of diverse sectors in Indonesia have further pushed the need for programming skills in areas, such as data analysis, artificial intelligence, big data, and other enterprise-level systems [4]. As the demand for digital solutions grows, maintaining the ability to adapt

popular programming languages to Indonesia's system has become more crucial as it can affect Indonesia's competitiveness in the world's digital landscape.

Globally, the popularity of programming languages can be tracked using the TIOBE Index. TIOBE Index is a rating of programming languages based on the number of skilled developers, courses, etc. worldwide. It is calculated at several popular sites, like Google, Bing, and Amazon. TIOBE Index can be used to determine the ranking of the programming language globally, thereby providing insights to the global and local developer about the current global trend of programming languages, helping them to make impactful decisions about which languages are suitable for their future projects.

However, global trends are not necessarily the same as local trends as challenges exist in adopting a particular programming language in Indonesia. One example is C++ which is popular globally but unfavored in Indonesia [5]. Another example is Javascript, which is the most famous language in Indonesia due to the local trend of using website applications for promotion and other tasks. Moreover, different sectors may have unique challenges and issues, leading to distinct preferences for programming languages and diverse effects that the programming language has on that sector.

The domain of analyzing the correlation of global popularity ratings of programming languages with the job market and search trends in Indonesia is crucial for tech industries and education institutions. With statistical analysis, we aim to compare the significance of the correlation between each factor and trends between old and new programming languages. The result can give insights to developers in tech and educational companies to further determine their decisions in making certain projects or curricula for students. Understanding the dynamics between the trends can help us explore how the global trends of programming language can shape the sector technology landscape in Indonesia.

B. HYPOTHESIS

First Section: Comparing Significance of Correlation Between Each Factor

Because there are 8 factors in our dataset, we would not like to test all of the 56 hypotheses. Therefore, we will first visualize the Pearson correlation heatmap and define strong relations of two factors if they have a Pearson coefficient greater than or equal to 0.7 and lesser than or equal to -0.7.

After that, factors that have significant linear relations based on the Pearson correlation heatmap will be tested further using the Chi-test. The general hypothesis will be

H0: Factor A is not significantly correlated to factor B

H1: Factor A is significantly correlated to factor B

Second Section: Comparing Old and New Programming Languages

Part 1: Comparing Google Trends between old and new programming languages

H0: The average Google Trends scores for newer languages are equal to older languages

H1: The average Google Trends scores for newer languages differ from older languages

Part 2: Comparing Wikipedia views between old and new programming languages

H0: The average Wikipedia views for newer languages are equal to older languages

H1: The average Wikipedia views for newer languages differ from older languages

C. DATASET

Since we aim to predict the impacts of programming language global ratings on job and search trends, we need datasets that align with them. We also need the global popularity ratings datasets to determine the rankings of each programming language. We decided to use the TIOBE index as our benchmark to determine the global popularity ratings. For the other datasets, because we need local (Indonesian) datasets, we mainly scrap the web to acquire them. Below is the list of datasets we use:

- Programming language list: Kaggle repository
- Programming language rating: TIOBE Index
- Job datasets: total job listings, salary, and location from Indeed and Jobstreet, number of people who state the programming language in their skills in LinkedIn.
- Search trends datasets: Wikipedia Pageviews Analysis, Google Trends, Stack Overflow Developer Survey, GitHub users search

Below is each dataset's details and preprocessing method:

a. Programming Language Name List

The initial step that we do is finding a dataset of all the programming languages in this world. We first found a Kaggle dataset by Sujay Kapadnis[1] which contains names of programming languages and their description, type, and so on. However, after further inspection, the repository is sourced from a more complete programming language repository by the same author[2]. The dataset contains about 4000 programming language names and 135000 facts about them (its description, features, etc). Upon further investigation, the datasets are obtained from pldb.io[3]. Pldb.io is a public-domain scroll set and website that serves as a database for programming languages. Its sources can be seen on their acknowledgment page[4].

After acquiring the programming language dataset, we dropped all the invalid records. For example, the OS Android is also inside the dataset. So, we dropped the records that have the invalid type (not a programming language). At first, we wanted to use the data (GitHub repos, Stack Overflow, etc.) inside the dataset. However, because the data is not local, we decided not to use it. Finally, we convert the list of the names of the programming languages into a CSV file.

b. Job Dataset

To acquire the job dataset, we did web scraping from id.indeed and jobstreet Indonesia which are both popular job-searching sites. The raw scrapped CSV files from id.indeed have six attributes, i.e. job title, company name, location, salary, job type, and job description. Meanwhile, raw CSV files from jobstreet contain seven attributes, i.e. job title, company name, location, salary, job type, job description, and date. Then, we decided to drop the date attribute from the jobstreet CSV files since it has many NaN values and we need to uniform the acquired CSV files' attributes.

For the preprocessing method, we fill the NaN values in the salary attributes using the mean strategy in the SimpleImputer class. Then, we concatenate all files and remove the duplicates. After that, we clean the job title and extract the programming language from the job description. Lastly, we remove the possible outliers from the datasets.

[1] See more at https://www.kaggle.com/datasets/sujaykapadnis/programming-languages
[2] See more at https://www.kaggle.com/datasets/sujaykapadnis/programming-language-database
[3] See more at https://github.com/breck7/pldb
[4] See more at https://pldb.io/pages/acknowledgements.html

Next, we aim to transform the CSV file before to a new CSV file with the programming language name as the key attribute. So, we calculate the job amount and average salary based on each programming language. We also extract the job location into the new CSV file. The final result is then stored in job_result.csv.

c. Wikipedia Monthly Page Views Dataset

To determine the trend of a specific programming language, we decided to check for their monthly Wikipedia page views in Wikipedia Pageviews Analysis[5]. However, not all programming languages have their own Indonesian Wikipedia page, so we only record those who have it. We combined all the data of individual programming languages into a single dataset that contains the monthly page view from July 2015 to September 2024. After that, we calculate the average monthly views for each programming language, starting from the first month when the view is not 0 (when the page is first created). The result is stored in wiki_complete.csv.

d. Stack Overflow Annual Developer Survey Dataset

Besides Wikipedia Page monthly views, we also search for languages that Stack Overflow users (developers) utilize or desire to learn. We obtain the data from the Stack Overflow Insights[6] and clean it further into a single dataset by dropping all rows with country attributes not equal to 'Indonesia' and dropping unrelated columns. Then, for users who do not fill in the information about the languages that they have used or desired, their records are also dropped. After that, we count the occurrences of each language in the column. The result is in stackOverflow_complete.csv.

e. Other Datasets

For other datasets, we use the sweat equity strategy to gather the data. These datasets include the GitHub users, Google Trends, TIOBE index rating, and LinkedIn skill datasets. In other words, we gather the data manually for these datasets. We search and record the search results in the CSV files.

After acquiring all datasets, we merge them into a CSV file called merge.csv. The total language we got is 175, which each consists of ten attributes as below:

[5] See more at https://pageviews.wmcloud.org/
[6] See more at https://survey.stackoverflow.co/

Attribute Name	Description	Data Type
programming language	The name of the programming language	object (string)
tiobe index ratings	The rating of the programming language based on the TIOBE index	float
job amount	The number of jobs based on the programming language	int
average salary	The average salary of jobs in a specific programming language	float
location	The location (province) of programming language jobs	object (list)
linkedin skill	The number of people who put the programming language as their skill on LinkedIn	int
avg wiki views (monthly)	The average of Wikipedia views on the programming language page monthly	float
github user count	The number of GitHub users who used the programming language in their repository	int
average search count	The average number of searches done by people monthly	float
stack overflow count	The number of Stack Overflow users that desire or have worked with the specific programming language	int

To make it clearer, below is the dataset snippet:

programming language	tiobe index ratings	job amount	average salary	location	linkedin skill	avg wiki views (monthly)	github user count	average search count	stack overflow count
Java	10.51	728	6.153528e+06	['East Kalimantan', 'West Java', 'Bali', 'Yogy...	103000	3946.756757	6800	28.133930	2371
JavaScript	3.54	1111	6.168157e+06	['East Kalimantan', 'West Java', 'Bali', 'Yogy...	118000	5305.108108	11400	15.482143	4416
Dart	0.56	32	6.250161e+06	['West Java', 'Yogyakarta', 'East Java',	945	51.125000	877	37.000000	957

We also got the time series data for Wikipedia and Google Trends. The details of each time series dataset can be seen below.

Data	Time Period	Programming Language Count*
Google search count (Google Trends)	July 2015 to October 2024 (Monthly)	55
Wikipedia search count	July 2015 to October 2024 (Monthly)	49

*the programming language count only specifies the programming languages that have at least a value that is greater than zero in its time series data

For all CSV files, you can refer to:
https://github.com/Ella-Raputri/FoDS-FinalProject/tree/main/data

Dataset Usage:

- First dataset (merge.csv): first section of hypothesis testing (all 6 parts) is in https://github.com/Ella-Raputri/FoDS-FinalProject/blob/main/data/merge.csv.
- Google Trends dataset: second section of hypothesis testing (first part) is in https://github.com/Ella-Raputri/FoDS-FinalProject/blob/main/data/googletrends_complete.csv.
- Wikipedia search dataset: second section of hypothesis testing (second part) is in https://github.com/Ella-Raputri/FoDS-FinalProject/blob/main/data/wiki_complete.csv

D. RESULT AND DISCUSSION

Data Visualization and Analysis

We plot the histogram for numerical columns in the merge.csv (the merged dataset) as shown in Figure 1. The numerical columns in the merge.csv consist of TIOBE Index ratings, job amount, average salary, number of people who display the programming language as a LinkedIn skill, GitHub user count, Wikipedia total views, search count (Google Trends), and number of stack overflow users.

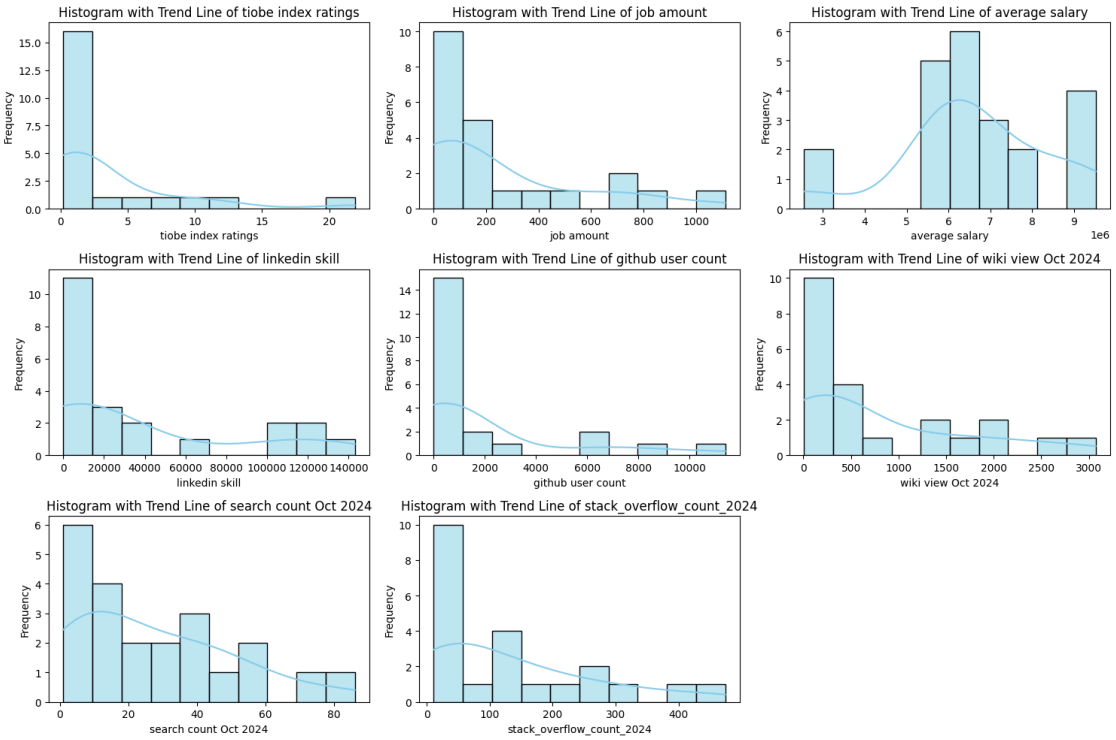

Figure 1: Histogram for Numerical Values in Merged Dataset

All of the histograms in Figure 1 are right-skewed, except for the average salary histogram which is left-skewed, but almost symmetric. We consider all isolated bars to be not outliers, but valid observations because super-popular programming languages like Python often have higher ratings, job amounts, user counts, etc. For example, the isolated bar in the TIOBE Index rating histogram is Python which has a rating above 20.

The histograms proved a big discrepancy between popular and unpopular programming languages regarding all factors, except the average salary. All the popular programming languages tend to become isolated bars or seem to be outliers due to their high numerical values. Meanwhile, unpopular programming languages dominate the lower numerical values.

The average salary histogram is the one that is distinct from the others due to its distribution shape. The reason is that Indonesian job salaries are not that different from those of mastering a certain programming language and another programming language. An unpopular programming language can also have a high average salary because if there is only one job that uses that language, then only the salary of that job will be the average salary of the unpopular programming language. One certain thing is that the average salary for programming or IT jobs in Indonesia is mostly about 5 to 7 million rupiah.

Next, we will also analyze the correlation of features in the merged programming language dataset by plotting the correlation matrix to observe the Pearson coefficient between each feature. We exclude the languages that do not have any LinkedIn skills, GitHub user count, etc. The result is shown below in Figure 2.

Figure 2: Pearson Correlation Matrix for Numerical Features in Merged Dataset

We define a coefficient greater than 0.7 and lesser than -0.7 as correlated. From all the features in the dataset, there are 6 relationships between features that have Pearson coefficients greater than 0.7, which are

- LinkedIn skill and job amount (0.91)
- GitHub user count and job amount (0.81)
- GitHub user count and LinkedIn skill (0.77)
- Stack Overflow count and job amount (0.89)
- Stack Overflow count and LinkedIn skill (0.85)
- Stack Overflow count and GitHub user count (0.75)

The heatmap shows that the previous 6 relationships have a significant linear correlation between the tested features, with the highest correlation being the relationship between LinkedIn skill and job amount. In other words, the more people that put the programming language as a skill in their LinkedIn, the more job requires that programming language and vice versa.

Upon further inspection, we can see that the relationship that has a high correlation coefficient is the permutation of two elements between these features: LinkedIn skill, job amount, GitHub user count, and Stack Overflow user count. All of the relationships between two features from those four features have a high correlation coefficient. This observation implies that the number of people who display the programming language as their skill in LinkedIn, job amount, GitHub user count, and Stack Overflow user count are all associated with each other. If one of them increases, then the others also increase. Conversely, the decrease in one of the four factors will highly likely make the other three values also decrease.

To better examine those 6 relations with the highest Pearson coefficient, we plot a scatter plot for each of the relations as shown below in Figure 3.

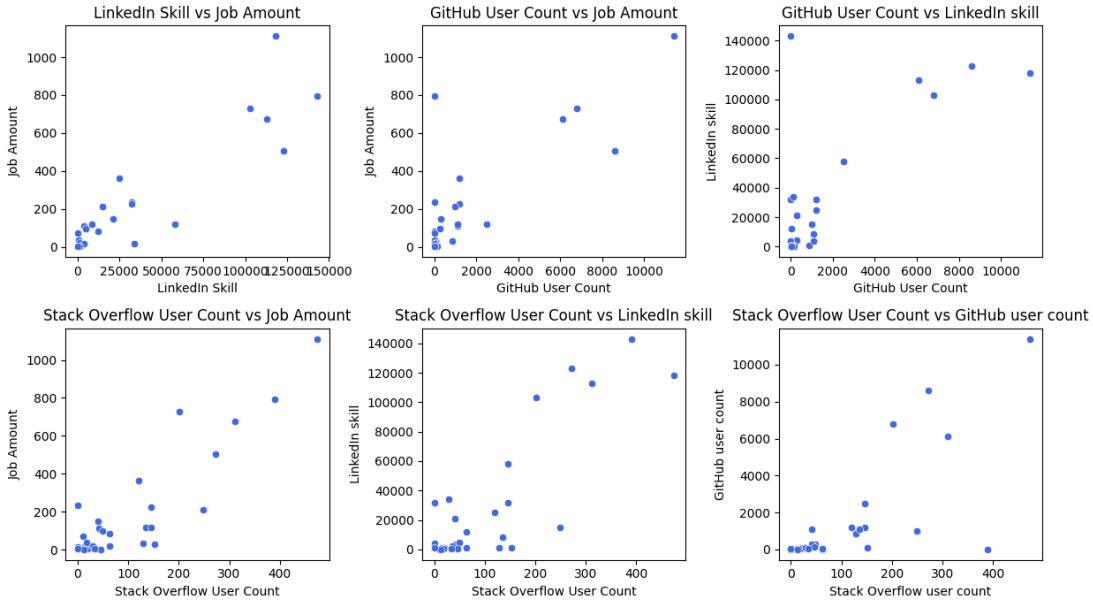

Figure 3: Scatter Plots for Relationship with Pearson Coefficient Greater than 0.7

Based on the scatter plots in Figure 3, all of the relations are not that linear as there are some data points that are far from the other data points that are similar for one of

the features. For example, in the GitHub user count versus job amount plot, we can see that even though there is a language that has a low GitHub user count, the job amount of that language is still considerably high.

Another similarity that we observed from the scatter plots is that the majority of the data points are in the lower value area. This observation further proved the discrepancy between unpopular and popular programming languages as discussed before in the histogram for numerical values.

Furthermore, the LinkedIn skill versus job amount scatter plot is unique as there are no middle-value data points, so there is a vast gap between the lower value data points with the higher value ones. Therefore, we can infer that based on the LinkedIn skill user amount and job amount, the data points can be categorized into low and high-value clusters. High-value clusters will be the popular programming languages, while the low-value clusters will consist of unpopular programming languages.

Scatter plots that plot the relation between GitHub user count with other factors (GitHub user count versus job amount, GitHub user count versus LinkedIn skill, and GitHub user count versus Stack Overflow user count) have outliers that are significantly different from the other data point. In GitHub user count versus job amount, there is a data point that has a low GitHub user count but a high job amount. In GitHub user count versus LinkedIn skill, there is a data point that has a low GitHub user count, but a high number of people displaying that language on LinkedIn. In GitHub user count versus Stack Overflow user count, there are several data points that have low GitHub user count, but high Stack Overflow user count. These anomalies may be the reason why the 3 relationships between GitHub user count with other factors are the lowest among the 6 relationships that have a Pearson coefficient greater than 0.7.

Meanwhile, for Stack Overflow user count versus job amount and Stack Overflow user count versus LinkedIn skill scatter plots, the linear relations are more visible. There exist low, middle, and high value data points. However, same as other scatter plots, the lower value dominates the amount of data points in the scatter plots.

Next, we also gathered insights regarding the programming languages' job market in Indonesia. Our research found that the most popular programming language in Indonesia's job market is JavaScript followed by SQL and Java. We also investigate each programming language's job type distribution as seen in Figure 4. As we can see from Figure 4, most programming languages have various job types distributions with the

developer type of job remaining as the most widely sought job. Since there are many types of developers of each programming language, thus we classify them under the label of “other developer”.

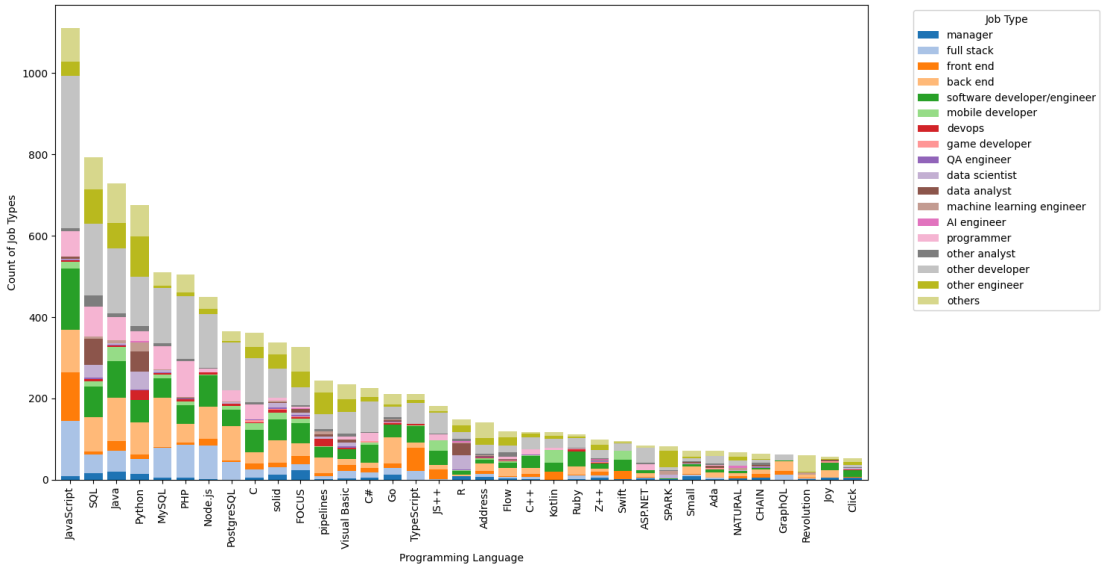

Figure 4: Job Type by Programming Language

We also investigated the job location of programming languages in Indonesia. As shown in Figure 5, most programming language job locations are in Jakarta followed by Banten and West Java. This result indicates that the programming job is more popular and sought on Java Island than on other big islands, such as Sumatra, Kalimantan, Sulawesi, or Papua, which can be interpreted that finding a programming job in Java, especially in Jakarta, is easier than the other places.

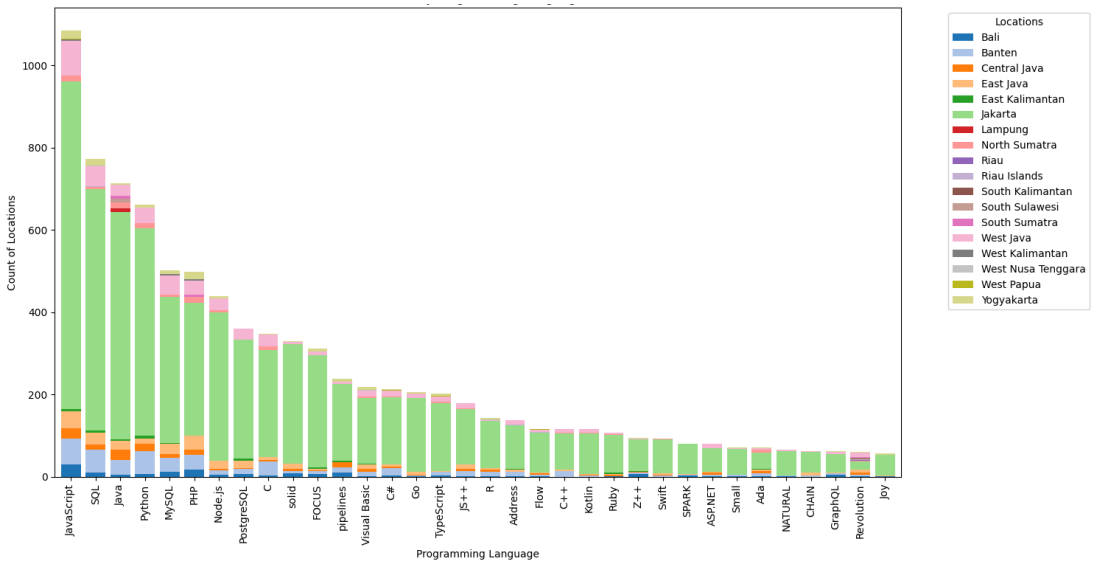

Figure 5: Location Count by Programming Language

Statistical Analysis and Hypothesis Testing

First Section: Comparing Significance of Correlation Between Each Factor

Based on the heatmap in the previous part, there are 6 relationships between features that have significant Pearson coefficients, which are

- LinkedIn skill and job amount
- GitHub user count and job amount
- GitHub user count and LinkedIn skill
- Stack Overflow count and job amount
- Stack Overflow count and LinkedIn skill
- Stack Overflow count and GitHub user count

Therefore, we would like to analyze them and determine whether the attributes are significantly dependent and affect each other, including the -1 values which are excluded in the heatmap. For each relation, we will conduct a hypothesis testing using Chi Test, so that we are sure that these 6 relations are also categorically dependent (including all values), besides linearly dependent for the known values (non -1 values) based on the Pearson heatmap.

Below are the categories of each factor:

- LinkedIn skill: Zero and Nonzero
- Job amount: Low and High
- GitHub user count: -1 (Language Not Available), Zero, and Nonzero
- Stack Overflow user count: Zero and Nonzero

To categorize the languages with low and high job amounts, we utilized the KMeans algorithm to create two clusters of the job amount and categorize each language to have low or high job amounts.

For each Chi test, we will first define the H0 and H1, then show the observed value in a table. After that, we calculate the expected value using this formula.

$$\text{Expected value} = \frac{\text{row total} \times \text{column total}}{\text{grand total}}$$

The result of each expected value will be shown in the expected value table. Then, we also calculate the $\frac{(O-E)^2}{E}$ with O as the observed value and E as the expected value. The result will then be shown in a table as well.

Lastly, we calculate the X^2 test statistic and df (degree of freedom). The test statistic X^2 is the sum of values in all cells in the last table. Meanwhile, df is the multiplication of the table's number of columns minus 1 with the table's number of rows minus 1. We then compare the X^2 with the critical value from the Chi-Square distribution table based on $\alpha = 0.05$ and the previous df.

Now, we will conduct the hypothesis testing.

1. **LinkedIn Skill Count and Job Amount**

   H0: LinkedIn skill count is not significantly correlated to job amount

   H1: LinkedIn skill count is significantly correlated to job amount

Observed Value Table for LinkedIn Skill and Job Amount

LinkedIn Skill	Low Job Amount	High Job Amount	Row Total
Zero	113	2	115
Nonzero	51	9	60
Column Total	164	11	175

Expected Value Table for LinkedIn Skill and Job Amount

LinkedIn Skill	Low Job Amount	High Job Amount	Row Total
Zero	107.77	7.23	115
Nonzero	56.23	3.77	60
Column Total	164	11	175

$\frac{(O-E)^2}{E}$ Value Table for LinkedIn Skill and Job Amount

LinkedIn Skill	Low Job Amount	High Job Amount

Zero	0.254	3.783
Nonzero	0.486	7.255

$X^2 = 0.254 + 3.783 + 0.486 + 7.255 = 11.778$

$df = (2 - 1)(2 - 1) = 1$

We will compare the X^2 statistic value with the critical value from the Chi-Square distribution table for df = 1 and $\alpha = 0.05$, which has a critical value of 3.841. Because $11.778 > 3.841$, then we can reject our null hypothesis because our obtained statistic is higher than the critical value.

2. GitHub User Count and Job Amount

H0: GitHub user count is not significantly correlated to the job amount

H1: GitHub user count is significantly correlated to the job amount

Observed Value Table for GitHub User Count and Job Amount

GitHub user count	Low Job Amount	High Job Amount	Row Total
-1	117	5	122
Zero	17	0	17
Nonzero	30	6	36
Column Total	164	11	175

Expected Value Table for GitHub User Count and Job Amount

GitHub user count	Low Job Amount	High Job Amount	Row Total
-1	114.33	7.67	122
Zero	15.93	1.06	17
Nonzero	33.73	2.26	36
Column Total	164	11	175

$\frac{(O-E)^2}{E}$ Value Table for GitHub User Count and Job Amount

GitHub user count	Low Job Amount	High Job Amount
-1	0.062	0.929
Zero	0.072	1.06
Nonzero	0.412	6.189

$X^2 = 0.062 + 0.929 + 0.072 + 1.06 + 0.412 + 6.189 = 8.724$

$df = (2 - 1)(3 - 1) = 2$

We will compare the X^2 statistic value with the critical value from the Chi-Square distribution table for df = 2 and $\alpha = 0.05$, which has a critical value of 5.991. Because $8.724 > 5.991$, then we can reject our null hypothesis because our obtained statistic is higher than the critical value.

3. GitHub User Count and LinkedIn Skill Count

H0: GitHub user count is not significantly correlated to amount of people display the programming language in their skill section in LinkedIn

H1: GitHub user count is significantly correlated to amount of people display the programming language in their skill section in LinkedIn

Observed Value Table for GitHub User Count and LinkedIn Skill Count

GitHub user count	Zero LinkedIn	Nonzero LinkedIn	Row Total
-1	99	23	122
Zero	12	5	17
Nonzero	4	32	36
Column Total	115	60	175

Expected Value Table for GitHub User Count and LinkedIn Skill Count

GitHub user count	Zero LinkedIn	Nonzero LinkedIn	Row Total
-1	80.17	41.83	122
Zero	11.17	5.83	17

Nonzero	23.66	12.34	36
Column Total	115	60	175

$\frac{(O-E)^2}{E}$ **Value Table for GitHub User Count and LinkedIn Skill Count**

GitHub user count	Zero LinkedIn	Nonzero LinkedIn
-1	4.422	8.476
Zero	0.062	0.118
Nonzero	16.336	31.322

$X^2 = 4.422 + 8.476 + 0.062 + 0.118 + 16.336 + 31.322 = 60.736$

$df = (2-1)(3-1) = 2$

We will compare the X^2 statistic value with the critical value from the Chi-Square distribution table for df = 2 and $\alpha = 0.05$, which has a critical value of 5.991. Because $60.736 > 5.991$, then we can reject our null hypothesis because our obtained statistic is higher than the critical value.

4. **Stack Overflow User Count and Job Amount**

H0: Stack Overflow user count is not significantly correlated to job amount

H1: Stack Overflow user count is significantly correlated to job amount

Observed Value Table for Stack Overflow User Count and Job Amount

Stack Overflow User	Low Job Amount	High Job Amount	Row Total
Zero	132	2	134
Nonzero	32	9	41
Column Total	164	11	175

Expected Value Table for Stack Overflow User Count and Job Amount

Stack Overflow User	Low Job Amount	High Job Amount	Row Total

Zero	125.58	8.42	134
Nonzero	38.42	2.57	41
Column Total	164	11	175

$\frac{(O-E)^2}{E}$ Value Table for Stack Overflow User Count and Job Amount

Stack Overflow User	Low Job Amount	High Job Amount
Zero	0.328	4.895
Nonzero	1.073	16.087

$X^2 = 0.328 + 4.895 + 1.073 + 16.087 = 22.383$

$df = (2-1)(2-1) = 1$

We will compare the X^2 statistic value with the critical value from the Chi-Square distribution table for df = 1 and $\alpha = 0.05$, which has a critical value of 3.841. Because $22.383 > 3.841$, then we can reject our null hypothesis because our obtained statistic is higher than the critical value.

5. Stack Overflow User Count and LinkedIn Skill Count

H0: Stack Overflow user count is not significantly correlated to LinkedIn skill user count

H1: Stack Overflow user count is significantly correlated to LinkedIn skill user count

Observed Value Table for Stack Overflow User Count and LinkedIn Skill Count

Stack Overflow User	Zero LinkedIn	Nonzero LinkedIn	Row Total
Zero	110	24	134
Nonzero	5	36	41
Column Total	115	60	175

Expected Value Table for Stack Overflow User Count and LinkedIn Skill Count

Stack Overflow User	Zero LinkedIn	Nonzero LinkedIn	Row Total
Zero	88.06	45.94	134
Nonzero	26.94	14.06	41
Column Total	115	60	175

$\frac{(O-E)^2}{E}$ **Value Table for Stack Overflow User Count and LinkedIn Skill Count**

Stack Overflow User	Zero LinkedIn	Nonzero LinkedIn
Zero	5.466	10.478
Nonzero	17.868	34.236

$X^2 = 5.466 + 10.478 + 17.868 + 34.236 = 68.048$

$df = (2-1)(2-1) = 1$

We will compare the X^2 statistic value with the critical value from the Chi-Square distribution table for df = 1 and $\alpha = 0.05$, which has a critical value of 3.841. Because $68.048 > 3.841$, then we can reject our null hypothesis because our obtained statistic is higher than the critical value.

6. **GitHub User Count and Stack Overflow User Count**

H0: GitHub user count is not significantly correlated to Stack Overflow user count.

H1: GitHub user count is significantly correlated to Stack Overflow user count.

Observed Value Table for GitHub User Count and Stack Overflow User Count

GitHub user count	Zero Stack Overflow	Nonzero Stack Overflow	Row Total
-1	113	9	122
Zero	16	1	17
Nonzero	5	31	36
Column Total	134	41	175

Expected Value Table for GitHub User Count and Stack Overflow User Count

GitHub user count	Zero Stack Overflow	Nonzero Stack Overflow	Row Total
-1	93.42	28.58	122
Zero	13.02	3.98	17
Nonzero	27.57	8.43	36
Column Total	134	41	175

$\frac{(O-E)^2}{E}$ **Value Table for GitHub User Count and Stack Overflow User Count**

GitHub user count	Zero Stack Overflow	Nonzero Stack Overflow
-1	4.103	13.414
Zero	0.682	2.231
Nonzero	18.477	60.428

$X^2 = 4.103 + 13.414 + 0.682 + 2.231 + 18.477 + 60.428 = 99.335$

$df = (2-1)(3-1) = 2$

We will compare the X^2 statistic value with the critical value from the Chi-Square distribution table for df = 2 and $\alpha = 0.05$, which has a critical value of 5.991. Because $99.335 > 5.991$, then we can reject our null hypothesis because our obtained statistic is higher than the critical value.

Second Section: Comparing Old and New Programming Languages

Because we have the Google Trends and Wikipedia data of all programming languages, we decided to test an interesting hypothesis about whether the newer languages' Google Trends search count and Wikipedia views differ significantly from the older languages. So, we first drop the zero columns in Wikipedia and Google Trends datasets. Then, we classify the programming languages into two groups, i.e. older and newer languages. Older languages can be defined as programming languages that were created before 2000.

We will use the Shapiro-Wilk Normality Test, Box-Cox transformation, Levene's test, and t-test with two independent samples. Shapiro-Wilk Normality Test tested the data distribution to determine whether it is normalized or not [6]. Box-Cox transformation is a parametric power transformation technique to reduce anomalies such as non-additivity, non-normality, and heteroscedasticity [7]. Levene's test is a powerful technique to test whether the data has equal variances [8]. For the Shapiro-Wilk Normality Test, Box-Cox transformation, and Levene's test, we used the Python built-in function (since we did not learn them in class). As for the α value, we used α=0.05 for all the tests.

a. Comparing Google Trends between old and new programming languages

   H0: The average Google Trends scores for newer languages are equal to older languages

   H1: The average Google Trends scores for newer languages differ from older languages

   We first test whether the data is normalized with the Shapiro-Wilk Normality Test. We got the result that newer languages' data has a p-value of 0.0002 and older languages' data has a p-value of 0.0151. Since their p-value is lower than 0.05, it is profound that their data is not normalized.

   Thus, we perform the Box-Cox transformation to normalize their data. After performing the Box-Cox transformation, we perform another Shapiro-Wilk Normality Test. We got that the p-value of newer languages' data is 0.4830 and older languages' data is 0.3128. Since both have a p-value greater than 0.05, then these data are already normalized.

   Next, we check whether they have equal variances. We use Levene's Test and acquire a p-value of 0.0867. Since the p-value is greater than 0.05, we assume that the variance is not significantly different.

   Since we already know that the data is normalized with equal variances, thus we perform a t-test for two independent samples assuming equal variances. For each group, we calculated the average, standard deviation, and sample size.

Group	Sample size (n)	Average ($\bar{x}$)	Standard deviation (s)
Old	31	6.1572	3.8874

New	24	2.2896	2.7453

We start by calculating the degree of freedom: 31 + 24 - 2 = 53. Then, we calculated the pooled variance to get the pooled standard deviation.

$$s_p{}^2 = \frac{((n_1-1){s_1}^2+(n_2-1){s_2}^2)}{n_1+n_2-2}$$
$$s_p{}^2 = \frac{((31-1)3.8874^2+(24-1)2.7453^2)}{31+24-2}$$
$$s_p{}^2 = \frac{(453.3564+173.3435)}{53}$$
$$s_p{}^2 = 11.8245$$
$$s_p = \sqrt{11.8245} = 3.4387$$

After that, we calculated the t-value. We assume the population mean difference (μ1 - μ2) is zero. We then got the t-value = 4.1365.

$$t = \frac{(x_1-x_2)-(\mu_1-\mu_2)}{s_p\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}}$$
$$t = \frac{(6.1572-2.2896)-0}{3.4387\sqrt{\frac{1}{31}+\frac{1}{24}}}$$
$$t = \frac{3.8676}{3.4387\times 0.2719} = 4.1365$$

Lastly, we looked at the t-table and got the table t-value of 2.0057. Since our calculated t-value is greater than the table t-value, thus we can reject the null hypothesis. So, we can infer that the average Google Trends scores for newer languages differ from older languages. The box plot for these two groups can be seen below:

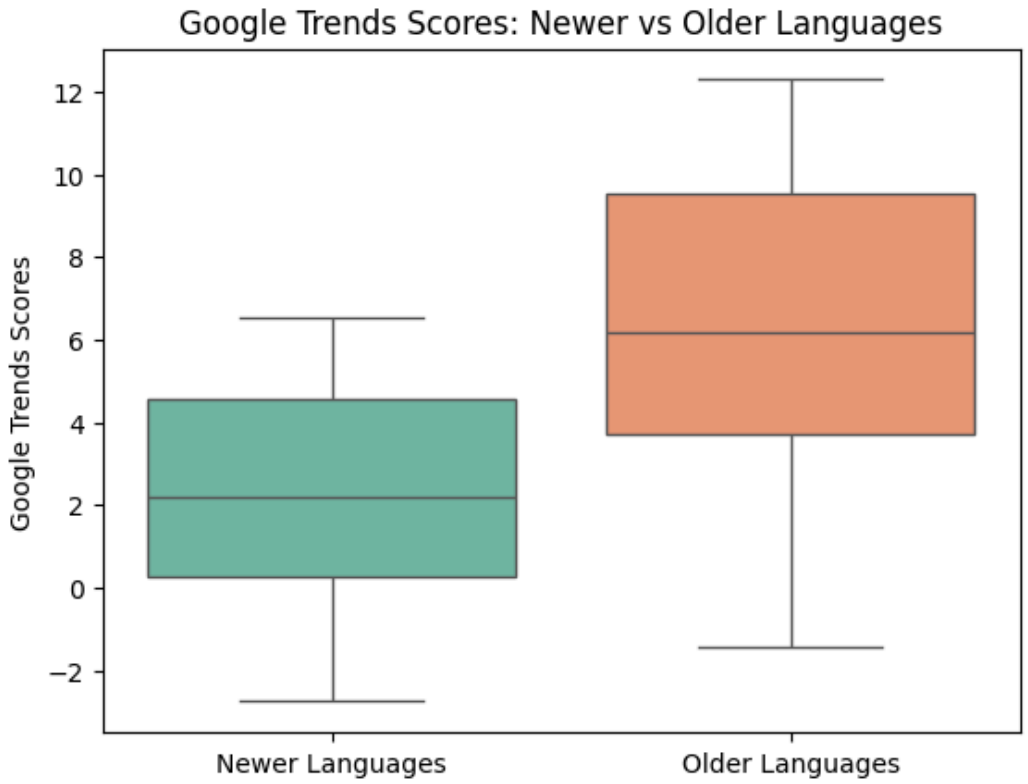

Figure 6: Older Languages vs Newer Languages Google Trends Score

b. Comparing Wikipedia views between old and new programming languages
H0: The average Wikipedia views for newer languages are equal to older languages
H1: The average Wikipedia views for newer languages differ from older languages

We first test whether the data is normalized with the Shapiro-Wilk Normality Test. We got the result that newer languages' data has a p-value of 2.7940×10^{-6} and older languages' data has a p-value of 1.4096×10^{-5}. Since their p-value is lower than 0.05, it is profound that their data is not normalized.

Thus, we perform the Box-Cox transformation to normalize their data. After performing the Box-Cox transformation, we got that the p-value of newer languages' data is 0.5715 and older languages' data is 0.3587. Since both have a p-value greater than 0.05, then these data are already normalized.

Next, we check whether they have equal variances. We use Levene's Test and acquire a p-value of 8.5218. Since the p-value is greater than 0.05, we assume that the variance is not significantly different.

Since we already know that the data is normalized with equal variances, thus we perform a t-test for two independent samples assuming equal variances. For each group, we calculated the average, standard deviation, and sample size.

Group	Sample size (n)	Average ($\bar{x}$)	Standard deviation (s)
Old	29	11.7738	4.9755
New	20	4.2764	1.0033

We start by calculating the degree of freedom: 29 + 20 - 2 = 47. Then, we calculated the pooled variance to get the pooled standard deviation.

$$s_p{}^2 = \frac{((n_1-1){s_1}^2+(n_2-1){s_2}^2)}{n_1+n_2-2}$$
$$s_p{}^2 = \frac{((29-1)4.9755^2+(20-1)1.0033^2)}{29+20-2}$$
$$s_p{}^2 = \frac{(693.1568+19.1256)}{47}$$
$$s_p{}^2 = 15.1549$$
$$s_p = \sqrt{15.1549} = 3.8929$$

After that, we calculated the t-value. We assume the population mean difference (μ1 - μ2) is zero. We then got the t-value = 6.6251.

$$t = \frac{(x_1-x_2)-(\mu_1-\mu_2)}{s_p\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}}$$
$$t = \frac{(11.7738-4.2764)-0}{3.8929\sqrt{\frac{1}{29}+\frac{1}{20}}}$$
$$t = \frac{7.4974}{3.8929\times 0.2907} = 6.6251$$

Lastly, we looked at the t-table and got the table t-value of 2.0117. Since our calculated t-value is greater than the table t-value, thus we can reject the null hypothesis. So, we can infer that the average Wikipedia views for newer languages differ from older languages. The box plot for these two groups can be seen below:

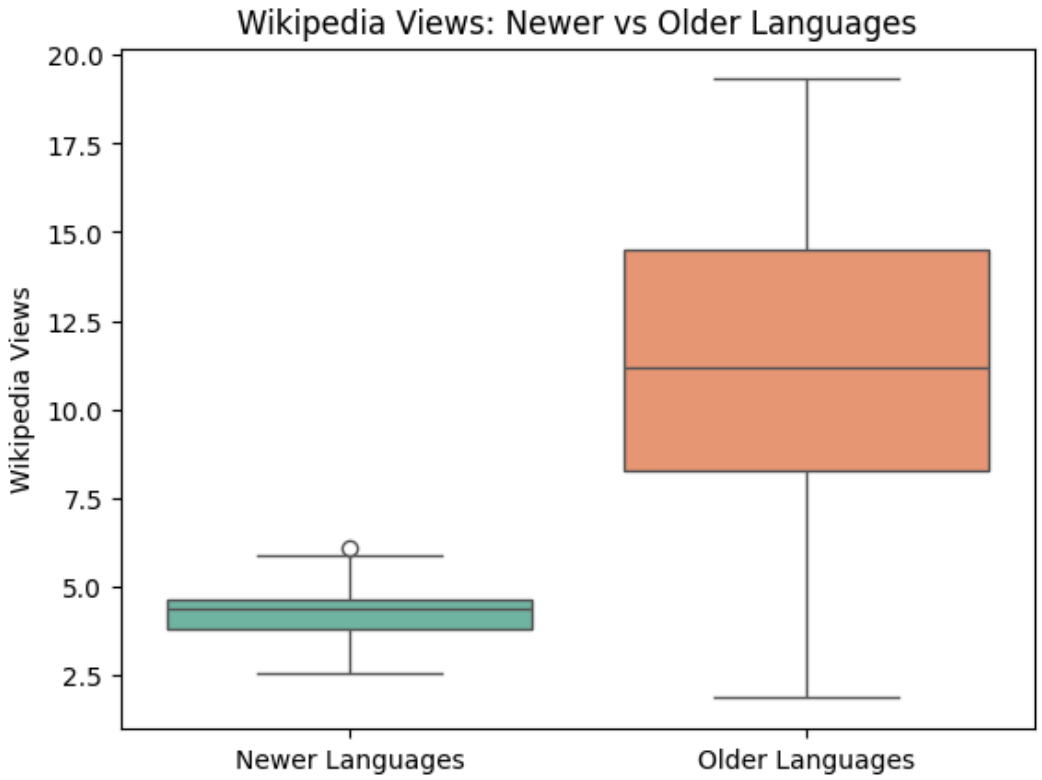

Figure 7: Older Languages vs Newer Languages Wikipedia Views Count

E. CONCLUSION

First Section: Comparing Significance of Correlation Between Each Factor

- LinkedIn skill count is significantly correlated to job amount
- GitHub user count is significantly correlated to job amount
- GitHub user count is significantly correlated to amount of people display the programming language in their skill section in LinkedIn
- Stack Overflow user count is significantly correlated to job amount
- Stack Overflow user count is significantly correlated to LinkedIn skill user count
- GitHub user count is significantly correlated to Stack Overflow user count.

Second Section: Comparing Old and New Programming Languages

Part 1: Comparing Google Trends between old and new programming languages

Conclusion: The average Google Trends scores for newer languages differ from older languages

Part 2: Comparing Wikipedia views between old and new programming languages

Conclusion: The average Wikipedia views for newer languages differ from older languages

REFERENCES

[1] D. Tošić, “Role of programming languages in digitalization,” in *Rev. NCD*, 2024, pp. 28–37.

[2] M. Shaw, “Myths and mythconceptions: what does it mean to be a programming language, anyhow?,” in *Proc. ACM Program. Lang.*, vol. 4, no. HOPL, pp. 1–44, Jun. 2020, doi: 10.1145/3480947.

[3] S. Pramana, “Peningkatan literasi data menuju Indonesia 4.0,” in *Empower. Comm.*, 2020.

[4] U. V. Wardina, N. Jalinus, and L. Asnur, “Kurikulum pendidikan vokasi pada era revolusi industri 4.0,” (in Indonesian), *J. Pend.*, vol. 20, no. 1, pp. 82–90, Mar. 2019, doi: 10.33830/jp.v20i1.240.2019.

[5] T. Indriyani, I. Arfyanti, M. Farkhan, I. N. A. Arsana, Joosten, and Sepriano, “Pengantar Bahasa Pemrograman Populer,” in *Bahasa Pemrograman Populer*, Jambi: Sonpedia, 2024.

[6] Z. Hanusz, J. Tarasinska, and W. Zielinski, “Shapiro–Wilk test with known mean,” *Revstat Stat. J*, vol. 14, no. 1, pp. 89–100, Feb. 2016, doi: https://doi.org/10.57805/revstat.v14i1.180.

[7] R. M. Sakia, “The Box-Cox transformation technique: A review,” *J. R. Stat.*, vol. 41, no. 2, p. 169, 1992, doi: https://doi.org/10.2307/2348250.

[8] J. L. Gastwirth, Y. R. Gel, and W. Miao, "The impact of Levene's test of equality of variances on statistical theory and practice," *Stat. Sci.*, pp. 343-360, 2009.