Jurnal TRANSFORMATIKA
Vol.21, No.1, Juli 2023, pp. 50-59
P-ISSN: 1693-3656, E-ISSN: 2460-6731
https://journals.usm.ac.id/index.php/transformatika/

Evaluating the Popularity of Programming Languages in Indonesia using the MABAC Method

Edi Widodo[1*], Rastri Prathivi[2], Soiful Hadi[3]
[1]Universitas Semarang
Jl.Sukarno Hatta Semarang, 024-6702757, e-mail: ediwidodo@usm.ac.id
[2]Universitas Semarang
Jl.Sukarno Hatta Semarang, 024-6702757, e-mail: vivi@usm.ac.id
[3]Universitas Semarang
Jl.Sukarno Hatta Semarang, 024-6702757, e-mail: saiful@usm.ac.id

ARTICLE INFO

History of the article:
Received 30 December 2010
Received in revised form 30 April 2011
Accepted 26 September 2012
Available online 8 October 2012

Keywords:

Programming Language; MABAC; Tiobe Index; Ranking.

*** Correspondence:**

Telpon:
+62 (024) 12345678

E-mail:
ediwidodo@usm.ac.id

ABSTRACT

In today's fast-paced digital era, the selection of a programming language plays a crucial role in the success of software development projects. This research aims to create an index of popularity for programming languages using the multi-attributive border approximation area comparison (MABAC) method. The study considers four data sources, including Jobstreet.Com, LinkedIn.Com, Google Trends, and Tiobe.com, to obtain the necessary information for evaluating the popularity of programming languages in Indonesia. The data range for this study is from May 1, 2020, until April 31, 2021. The results of the study indicate that the top ten programming languages in terms of popularity in Indonesia are Java, SQL, php, JavaScript, C, C++, python, C#, Visual Basic, and Assembly. The index can serve as a useful guide for strategic decision-making regarding the selection of programming languages for addressing the needs of the information technology market in Indonesia. The study's findings can be useful for software developers, IT professionals, and decision-makers in organizations who need to select a programming language for their software projects in Indonesia. The MABAC method used in this study can also be applied to other contexts for evaluating the popularity of programming languages.

INTRODUCTION

In today's digital era, the selection of a programming language is a crucial decision in software development projects. The popularity of programming languages can affect the recruitment of qualified developers, the efficiency of software development, and the successful implementation of software projects[1]. Therefore, understanding the popularity of programming languages is essential for software developers, IT professionals, and decision-makers in organizations.

Due to the limited time for study and the knowledge of lecturers in following the development of programming languages, students must choose several of the many existing programming languages, which are the most appropriate to be studied and taught to students, to adapt to the demands of the needs of experts in programming.

Programming languages that are on the rise cannot always be used as an option in preparing college graduates to answer the needs of the workforce. Employment requires experts who are ready to support their business operations and who may already be using certain technologies. Therefore, one of the teaching objectives in the Indonesian National Qualifications Framework or abbreviated as KKNI must meet the work structure in various. Considering the needs of the jobs market. However, there is a lack of comprehensive and accurate information about the popularity of programming languages in specific regions and periods. For example, Indonesia is a developing country that is experiencing rapid growth in its information technology sector. However, there is a scarcity of studies that investigate the popularity of programming languages in Indonesia. Additionally, the popularity of programming languages can fluctuate due to technological advancements, changing business needs, and economic conditions. Therefore, it is necessary to have up-to-date and reliable data to assess the popularity of programming languages[2].

TIOBE stands for "The Importance of Being Earnest" a company that focuses on the assessment and tracking of software quality. Periodically issuing a ranking of the popularity of programming languages around the world using the tracking method through 25 online search engine sites using the keyword + “<language> programming”, this method has received a lot of criticism from information technology observers because it is considered very weak and still needs weighting on each criterion in determining the ranking[3].

TIOBE is a company specializing in assessing the quality of software. TIOBE's core technology is based on an elegant compiler from Philips Tiobe index is a popularity order of the 20 most popular programming languages issued by tiobe periodically every month, this order is based on the calculation results of rankings from 25 search web engines with a reach of all users in the world using the keyword +"<language> Programming."

RESEARCH METHODS

1. Theoretical Framework

Based on the existing problem background and previous research findings, this study hypothesizes that the programming language index issued by TIOBE will be significantly different, if the research subject is more focused on programming language users in Indonesia using different weighting methods. This study collected data from several websites that are considered representative of the programming language user community in Indonesia during the period from May 1, 2020, to April 31, 2021. The data obtained was then analyzed using the MABAC method to generate a new ranking of programming languages that is more suitable for the Indonesian community, with weighting based on several predetermined criteria.

The Multi-Attribute Border Approximation Area Comparison (MABAC) method is a decision-making method. Pamuara and Cirovic developed the MABAC method in 2015. This method is used for ranking several alternative options. The basic assumption of the MABAC method is reflected in the definition of the criterion function distance from each observed alternative in the boundary approximation region[4].

The MABAC method involves several steps in its calculation. These steps are as[5][6]:

a. Criteria identification: The first step involves identifying the criteria or attributes that are relevant for the decision-making process.
b. Weight assignment: The third step involves assigning weights to the criteria. This is done to reflect their relative importance in the decision-making process.

c. Normalization of criteria: The second step involves normalizing the criteria to bring them to a common scale. This is done to avoid any bias towards a particular criterion.
d. Construction of the decision matrix (X): The fourth step involves constructing a decision matrix that contains the normalized scores and weight assigned to each criterion for each alternative.

$$X = \begin{array}{c} \\ A_1 \\ A_2 \\ \cdots \\ A_m \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \cdots & C_3 \end{array} \\ \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{13} \\ x_{21} & x_{22} & \cdots & x_{23} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} \end{array} \quad (1)$$

where m is a number of alternative and n is number of criteria.

e. Determination of the ideal solution: The fifth step involves determining the ideal solution by finding the maximum score for each criterion.

$$t_{ij} = \frac{x_{ij} - x_i^-}{x_i^+ - x_i^-} \quad (3)$$

f. Determination of the negative ideal solution: The sixth step involves determining the negative ideal solution by finding the minimum score for each criterion

$$t_{ij} = \frac{x_{ij} - x_i^-}{x_i^+ - x_i^-} \quad (4)$$

where x_{ij} is matrix first decision matrix, x_i^- is $min(x_1, x_2, x_m)$ and $x_i^+$is $max(x_1, x_2, x_m)$

g. Calculation of separation measures (v): The seventh step involves calculating the separation measures for each alternative, which indicate the distance of each alternative from the ideal and negative ideal solutions.

$$V = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{13} \\ v_{21} & v_{22} & \cdots & v_{23} \\ \cdots & \cdots & \cdots & \cdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{pmatrix} \quad (5)$$

v_{ij} calculate with

$$v_{ij} = \left(w_i * t_{ij}\right) + w_i \quad (6)$$

where w_i is normalized matrix elemens and t_{ij} is criteria weight coefficient.

h. Determination of the boundary approximation area matrix (G) obtained a weighted matrix as follows:

$$V = \begin{pmatrix} (w_1 * t_{11}) + w_1 & (w_2 * t_{12}) + w_2 & \cdots & (w_n * t_{1n}) + w_n \\ (w_1 * t_{21}) + w_1 & (w_2 * t_{22}) + w_2 & \cdots & (w_n * t_{2n}) + w_n \\ \cdots & \cdots & \cdots & \cdots \\ (w_1 * t_{m1}) + w_1 & (w_2 * t_{m2}) + w_2 & \cdots & (w_n * t_{mn}) + w_n \end{pmatrix} \quad (7)$$

where n is number of total criteria and m is number of alternative

$$g_i = \left(\prod_{j=1}^{m} v_{ij}\right)^{1/m} \quad (8)$$

Where v_{ij} shows the weighted matrix element (V) and "m" represents the total number of alternatives. After calculating the values of g_i based on the criteria, form the boundary approximation area matrix G using formula (9) in the form of n x 1.

$$G = \begin{matrix} C_1 & C_2 & \cdots & C_3 \\ [g_1 & g_2 & \cdots & g_3] \end{matrix} \quad (9)$$

i. Calculation of alternative distance matrix elements from the boundary approximation area (Q).

$$Q = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ q_{m1} & q_{m2} & \cdots & q_{mn} \end{bmatrix} \quad (10)$$

Alternative distance from the estimated boundary area (q_{ij}) is determined as the difference between the weighted matrix elements (V) and the value of the boundary approximation area (G).

$$Q = V - G \quad (11)$$

$$Q = \begin{bmatrix} v_{11} - g_1 & v_{12} - g_2 & \cdots & v_{1n} - g_n \\ v_{21} - g_1 & v_{22} - g_2 & \cdots & v_{2n} - g_n \\ \cdots & \cdots & \cdots & \cdots \\ v_{m1} - g_1 & v_{m2} - g_2 & \cdots & v_{mn} - g_n \end{bmatrix} \quad (12)$$

Where g_i represents the boundary approximation area for criterion C_i, v_{ij} represents the weighted matrix element (V), “n” represents the weighted matrix element (V), “m” represents the alternative number. Alternatif A_i can be included in the boundary approximation area (G), upper approximation area (G^+) or lower approximation area (G^-), that is, A_i the upper approximation area (G^+) represents the area where the ideal alternative is located (A^+), while the lower approximation area (G^-) represents the area where the anti-ideal alternative is located (A^-). The inclusion of alternative A^i in the approximation (G , G^+, G^-) is determined based on formula (13).

$$A_i \in \begin{Bmatrix} G^+ \; if \; q_{ij} > 0 \\ G \quad if \; q_{ij} = 0 \\ G^- \; if \; q_{ij} < 0 \end{Bmatrix} \quad (13)$$

To be selected as the best from the set, alternative A_i must be included in the upper approximation area (G^+) with as many criteria as possible.

j. Ranking of alternatives. The calculation of criterion function values with alternative (14) is obtained as the sum of alternative distance from the boundary approximation area (q_i). By adding the elements of the Q matrix with lines, the final value of the alternative criterion function is obtained.

$$S_i = \sum_{j=1}^{n} q_{ij} \, , j = 1,2, \ldots , n \; , i = 1,2, \ldots , m \quad (14)$$

where n is the number of criteria and m is the number of alternatives.

Multi-Criteria Decision Making (MCDM) is a useful process for identifying the optimal alternative from a set of feasible options based on multiple criteria or attributes[7]. In order to develop a multi-criteria decision-making model for selecting the best programming language, it is important to conduct a detailed analysis of the criteria on which the evaluation will be based. This can help to ensure that the model accurately reflects the needs and priorities

DOI : 10.26623/transformatika.v21i2.7001

of the stakeholders involved and can facilitate more effective decision making in the context of the Indonesian information technology market.

Various methods have been used to assess the popularity of programming languages, including surveys, web searches, and website analytics. However, these methods have limitations in terms of scope, accuracy, and timeliness[1]. Therefore, researchers have proposed alternative methods, such as the MABAC method, which can address these limitations[4].

Overall, the MABAC method has shown promise in providing a comprehensive and accurate assessment of the popularity of programming languages. Therefore, this study will use the MABAC method to assess the popularity of programming languages in Indonesia.

Xue-Guo Xu's research is a significant step towards promoting environmentally conscious practices in the manufacturing industry. The traditional methods of supplier selection have not considered environmental factors, and the modification of the MABAC method by Xu is an innovative solution to address this issue. The use of complex criteria in supplier selection for the automotive industry is particularly crucial as the industry has a significant impact on the environment. The application of the extended MABAC method proposed by Xu can help identify the most environmentally friendly suppliers, which can potentially reduce the industry's negative impact on the environment. Overall, I think this research is commendable and can contribute to the development of a more sustainable manufacturing industry[8].

The research conducted by Stevic is highly relevant in the field of supplier selection for the construction industry. The use of a multi-criteria decision-making approach is a sound methodology to ensure that all relevant factors are considered in selecting the best supplier. By considering criteria such as material quality, price, delivery time, and reputation, Stevic's model ensures that the selected supplier will not only meet the construction company's needs but also be reliable and trustworthy. The use of the MABAC algorithm is also a smart choice, as it has been proven to be effective in ranking complex criteria. Overall, Stevic's research provides valuable insights for companies seeking to improve their supplier selection process[5].

In the field of programming, ranking different programming languages is a common practice that helps developers and organizations to make informed decisions. It's interesting to see those different methods, such as SMART and TIOBE, can produce different rankings, highlighting the importance of understanding the criteria and methods used in ranking[9].

2. Implementation of MABAC

a. In determining the criteria, various sources were considered, including Jobstreet.com, Tiobe.com, Trends.google.com, and Linkedin.com. Jobstreet.com is seen as relevant for identifying programming languages that are most in demand in the job market. Trends.google.com provides insights into the programming language interests of learners in Indonesia. Linkedin.com, on the other hand, represents professional communities who have worked with programming languages in relevant fields. Finally, Tiobe.com is a widely used site that serves as a reference for collecting programming language data.

b. Weight assigment.

Table 1. Weight of criteria

Code	**Criteria**	**Weight** ($\mathbf{w}_i'$)	**Note**
C1	Job street	80	Representing job market needs
C2	LinkedIn	75	Representing the number of professionals
C3	Google trends	60	Popularity level based on search keywords
C4	Tiobe indeks	55	The index released by tiobe
Sum of weight		270	

c. Normalization of Criteria.

Table 2. Normalized weight of criteria

Code	**Criteria**	**Weight** (w'_i)	**Normalized** (w_i)
C1	JobStreet	80	0,29630
C2	LinkedIn	75	0,27778
C3	GoogleTrends	60	0,22222
C4	Tiobe Indeks	55	0,20370

d. Alternative data collection method for choice. The choice options are obtained from the TIOBE Index, by taking the data of the top 10 most popular programming languages. If the keyword used consists of more than one word, the search is conducted by combining the words without spaces and using spaces between the two words that make up the programming language, enclosed in quotation marks[10].

Tabel 3. List of alternative

Code	**Alternative *(a)***
A1	C
A2	Python
A3	Java
A4	C++
A5	C#
A6	Visual Basic
A7	JavaScript
A8	Assembly language
A9	PHP
A10	SQL

e. Construction of the decision matrix (X)

Table 4. Decision matrix (x)

Code	**C1**	**C2**	**C3**	**C4**
A1	11	5.000	65,81	13,38
A2	23	49.000	60,96	11,87
A3	1.363	30.000	60,19	11,74
A4	781	40.000	60,63	7,81
A5	985	25.000	51,52	4,41
A6	19	32.000	60,08	4,02
A7	1.075	71.000	65,77	2,45
A8	3	2.500	38,69	2,43
A9	884	95.000	67,46	1,86
A10	1.338	100.000	55,90	1,71

f. Min-Max normalization of matrix

Table 5: Min / max value

DOI : 10.26623/transformatika.v21i2.7001

	JobStreet	LinkedIn	Google	Tiobe
Min	3	2.500	38,69	1,71
Max	1.363	100.000	67,46	13,38

Table 6. Normalized criteria

Code	C1	C2	C3	C4
A1	0.00588	0.64103	0.94265	1.00000
A2	0.01471	0.47692	0.77407	0.87061
A3	1.00000	0.28205	0.74731	0.85947
A4	0.57206	0.38462	0.76260	0.52271
A5	0.72206	0.23077	0.44595	0.23136
A6	0.01176	0.30256	0.74348	0.19794
A7	0.78824	0.70256	0.94126	0.06341
A8	-	-	-	0.06170
A9	0.64779	0.94872	1.00000	0.01285
A10	0.98162	1.00000	0.59819	-

g. Weighted Matrix (V).

Table 7. Weighted matrix

Code	C1	C2	C3	C4
A1	0.29804	0.45584	0.43170	0.40741
A2	0.30065	0.41026	0.39424	0.38105
A3	0.59259	0.35613	0.38829	0.37878
A4	0.46580	0.38462	0.39169	0.31018
A5	0.51024	0.34188	0.32132	0.25083
A6	0.29978	0.36182	0.38744	0.24403
A7	0.52985	0.47293	0.43139	0.21662
A8	0.29630	0.27778	0.22222	0.21627
A9	0.48824	0.54131	0.44444	0.20632
A10	0.58715	0.55556	0.35515	0.20370

h. Boundary approximation area matrix (G)

Table 8. Boundary approximation area

Border Area	Job Street	Linked In	Google	Tiobe
G	0.41987	0.40715	0.37052	0.27177

i. Calculation of alternative distance matrix elements from the boundary approximation area (Q).

Table 9. Distance matrix from boundary area

	C1	C2	C3	C4
A1	-0.12183	0.04869	0.06118	0.13563
A2	-0.11921	0.00311	0.02372	0.10928
A3	0.17273	-0.05102	0.01777	0.10701
A4	0.04593	-0.02253	0.02117	0.03841

A5	0.09037	-0.06527	-0.04919	-0.02094
A6	-0.12009	-0.04533	0.01692	-0.02775
A7	0.10998	0.06579	0.06087	-0.05515
A8	-0.12357	-0.12937	-0.14829	-0.05550
A9	0.06837	0.13416	0.07393	-0.06545
A10	0.16728	0.14841	-0.01536	-0.06807

j. Ranking of alternatives (S)

Table 10. Alternative distance from boundary

Code	**Alternatif**	**S**
A1	C	0.12368
A2	Python	0.01689
A3	Java	0.24648
A4	C++	0.08297
A5	C#	-0.04503
A6	Visual Basic	-0.17624
A7	JavaScript	0.18149
A8	Assembly language	-0.45674
A9	PHP	0.21101
A10	SQL	0.23225

The positive value indicates the level of proximity to being preferred.

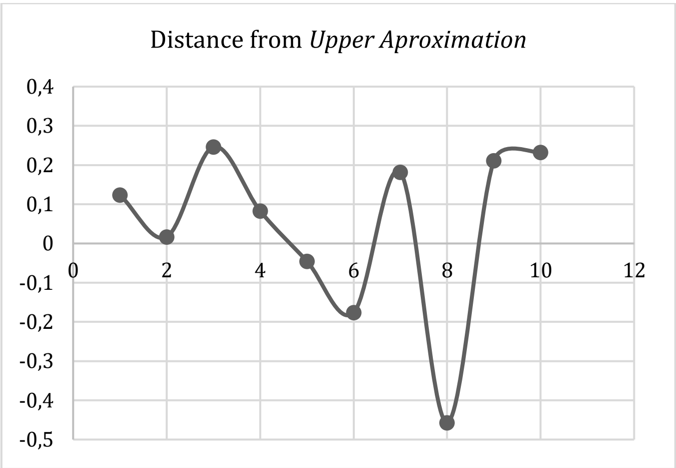

Figure 1: Alternative distance from upper

Table 11. Popularity rank based on MABAC.

Alternatives	**S**	**Rank**
Java	0.2464812	1
SQL	0.2322514	2
PHP	0.2110045	3
JavaScript	0.1814856	4
C	0.1236790	5

DOI : 10.26623/transformatika.v21i2.7001

C++	0.0829729	6
Python	0.0168900	7
C#	-0.0450323	8
Visual Basic	-0.1762361	9
Assembly language	-0.4567399	10

RESULTS

Based on the information provided, it seems that the research aims to utilize the MABAC method to evaluate the popularity of programming languages in the Indonesian market. The selection of criteria such as job web directories, LinkedIn, Google Trends, and Tiobe appears to be appropriate and may provide a comprehensive understanding of the popularity of programming languages in Indonesia. Additionally, the research's focus on data from only within Indonesia should help to provide a more localized perspective of the market.

The results of the research indicate that Java, SQL, PHP, JavaScript, C, C++, Python, C#, Visual Basic, and Assembly are the most popular programming languages in Indonesia, in that order.

DISCUSSION

These findings are consistent with those obtained using the SMART method in a previous study, which lends further support to the validity of the research's results. Overall, this research provides valuable insights into the current state of the Indonesian information technology market and can be used to inform strategic decision-making regarding the adoption of programming languages in the future.

CONCLUSIONS AND RECOMMENDATIONS

Determining popularity based solely on search results from specific online search engines, as done by the Tiobe method, cannot be considered a definitive measure of popularity as it does not adequately represent the needs of the workforce for the required programming languages. More important criteria include the number of users who have used a programming language for work, as well as the level of demand for programmers with specific programming language skills to support existing businesses.

Java remains the most popular programming language in the job market in Indonesia, with a high number of professionals continuing to work with this language due to its platform-independent nature. However, it should be noted that this research only utilized one job search directory. To improve the credibility of the obtained results, it would be beneficial to use multiple job search directories to increase objectivity and ensure better alignment with workforce needs.

REFERENCES

[1] Bissyande TF, Thung F, Lo D, Jiang L, Reveillere L. Popularity, interoperability, and impact of programming languages in 100,000 open-source projects. Proceedings - International Computer Software and Applications Conference, IEEE Computer Society; 2013, p. 303–12. https://doi.org/10.1109/COMPSAC.2013.55.

[2] Barua A, Thomas SW, Hassan AE. What are developers talking about? An analysis of topics and trends in Stack Overflow. Empirical Software Engineering 2014; 19:619–54. https://doi.org/10.1007/s10664-012-9231-y.

[3] Swaroop. How accurate are the language ratings published in the TIOBE index. Quora 2021. https://qr.ae/pGBw16 (accessed November 18, 2021).

[4] Alinezhad A, Javad K. New Methods and Applications in Multiple Attribute Decision Making (MADM). MABAC Method. In: New Methods and Applications in Multiple Attribute Decision Making (MADM). International Series in Operations Research & Management Science, vol 277, Springer, Cam.; 2019, p. 193–8. https://doi.org/10.1007/978-3-030-15009-9_25.
[5] Stević Ž, Pamučar D, Vasiljević M, Stojić G, Korica S. Novel integrated multi-criteria model for supplier selection: Case study construction company. Symmetry (Basel) 2017;9. https://doi.org/10.3390/sym9110279.
[6] Wang J, Wei G, Wei C, Wei Y. MABAC method for multiple attribute group decision making under q-rung orthopair fuzzy environment. Defence Technology 2020; 16:208–16. https://doi.org/10.1016/j.dt.2019.06.019.
[7] Pamučar D, Stević Ž, Zavadskas EK. Integration of interval rough AHP and interval rough MABAC methods for evaluating university web pages. Applied Soft Computing Journal 2018; 67:141–63. https://doi.org/10.1016/j.asoc.2018.02.057.
[8] Xue-Guo X, Shi H, Li-Jun Z, Hu-Chen L. Green supplier evaluation and selection with an extended MABAC method under the heterogeneous information environment. Sustainability (Switzerland) 2019;11. https://doi.org/10.3390/su11236616.
[9] Widodo E, Hadi S. Analisis Popularitas Bahasa Pemrograman Menggunakan Metode Simpe Multi Attribure Rating Technique (SMART) Untuk Menentukan Pilihan Dalam Mempelajari Bahasa Pemrograman. E-PROSIDING SEMINAR NASIONAL HASIL PENELITIAN LPPM UNIVERSITAS SEMARANG 2021:301–5.
[10] Wdzięczna D. SQL Pattern Matching 2017. https://learnsql.com/blog/using-like-match-patterns-sql/ (accessed June 7, 2021).

DOI : 10.26623/transformatika.v21i2.7001