

Keterbacaan kode memengaruhi pemahaman, beban kognitif pengembang, serta efektivitas pemeliharaan dan evolusi perangkat lunak. Keterbacaan kode bersifat multi dimensional dan subjektif, serta berada pada suatu spektrum tingkat kemudahan untuk dibaca. Oleh karena itu, keterbacaan kode tidak memadai apabila hanya direpresentasikan secara dikotomis sebagai *readable* atau *unreadable*. Diperlukan skor kontinu agar tingkat keterbacaan dapat dinyatakan sebagai derajat kualitas yang terukur. Representasi ini juga lebih fleksibel untuk diterapkan dalam analisis, pemeliharaan, dan pencarian kode. Skor keterbacaan kontinu berpotensi dimanfaatkan sebagai sinyal tambahan dalam pemeringkatan pada *code search*. Namun sebagian besar pendekatan *code search* saat ini masih mengutamakan relevansi leksikal dan semantik tanpa memasukan skor keterbacaan sebagai kriteria pemeringkatan. Penelitian ini bertujuan mengembangkan model keterbacaan kode multidimensi yang dapat menghasilkan skor kuantitatif sekaligus mengkaji pemanfaatannya sebagai faktor tambahan dalam pemeringkatan hasil *code search*.

Penelitian ini mengembangkan sebuah model keterbacaan kode yang memandang keterbacaan sebagai konstruk laten multidimensi yang terdiri atas lima dimensi, yaitu *Visual Clarity*, *Structural Simplicity*, *Documentation Support*, *Naming Transparency*, dan *Cognitive Load*. Model ini diberi nama Kerta. Model dibangun menggunakan 57 fitur yang diturunkan dari sintesis metrik keterbacaan dari penelitian sebelumnya dan dievaluasi menggunakan korpus berisi 2.482 cuplikan kode Java berlabel tiga kelas keterbacaan (*unreadable*, *neutral*, dan *readable*). Hasil evaluasi klasifikasi menunjukkan bahwa model Kerta mencapai kinerja *weighted F1* lebih baik ($wF1 = 0,86117$) dibanding model sebelumnya yaitu Scalabrino ($wF1 = 0,83537$), Buse-Weimer ($wF1 = 0,84386$), dan model gabungan ($wF1 = 0,85147$). Model Kerta juga menghasilkan skor keterbacaan yang memiliki hubungan monotonik kuat terhadap urutan label keterbacaan (Spearman $\rho = 0,797$), serta mampu membedakan tingkat keterbacaan secara lebih baik dibandingkan model keterbacaan sebelumnya (Scalabrino $\rho = 0,5256$ dan Buse-Weimer $\rho = 0,3075$). Analisis PLS-SEM menunjukkan bahwa kelima dimensi tersebut mampu menjelaskan 63,5% variansi keterbacaan, dengan *Naming Transparency* sebagai dimensi yang memberikan kontribusi paling dominan.

Selanjutnya, skor keterbacaan Kerta diintegrasikan ke dalam sistem *code search* berbasis BM25 dan model hibrida BM25 + *dense vector* menggunakan korpus *CodeSearchNet Challenge* Java. Hasil eksperimen menunjukkan bahwa skor keterbacaan berfungsi sebagai sinyal sekunder yang meningkatkan kualitas pemeringkatan tanpa mengganggu struktur ranking utama. Integrasi keterbacaan pada BM25 menghasilkan peningkatan kinerja yang konsisten sebesar +1,99% daripada *baseline* MMR=0,487. Sedangkan kombinasi relevansi leksikal, representasi semantik, dan skor keterbacaan menghasilkan model hibrida dengan kinerja terbaik (MRR@10 = 0,533). Temuan ini menunjukkan bahwa keterbacaan kode yang dimodelkan secara multidimensi dapat diukur secara efektif dan dimanfaatkan untuk meningkatkan kualitas hasil *code search*.

Kata kunci—Keterbacaan Kode Program; Pencarian Kode Program; Dimensi Konseptual Keterbacaan Kode; Klasifikasi Multi Kelas; Metrik Kerta; Perangkingan Berbasis Keterbacaan; Java

ABSTRACT

Code readability affects program comprehension, developers' cognitive load, and the effectiveness of software maintenance and evolution. It is multidimensional and subjective and exists along a continuum of ease of reading. Therefore, representing code readability solely as a dichotomy between *readable* and *unreadable* is inadequate. A continuous score is required to express readability as a measurable degree of quality. Such a representation is also more flexible for use in code analysis, maintenance, and search. A continuous readability score may serve as an additional ranking signal in *code search*. However, most existing *code search* approaches continue to prioritize lexical and semantic relevance without incorporating readability scores as a ranking criterion. This study therefore aims to develop a multidimensional code readability model that produces quantitative scores and to investigate their use as an additional factor in ranking *code search* results.

This study develops a code readability model that conceptualizes readability as a multidimensional latent construct comprising five dimensions: *Visual Clarity*, *Structural Simplicity*, *Documentation Support*, *Naming Transparency*, and *Cognitive Load*. The model is named Kerta. It was constructed using 57 features derived from a synthesis of readability metrics proposed in previous studies and evaluated on a corpus of 2,482 Java code snippets labeled into three readability classes: *unreadable*, *neutral*, and *readable*. The classification results show that Kerta achieved a higher weighted F1-score ($wF1 = 0.86117$) than the existing Scalabrino ($wF1 = 0.83537$), Buse–Weimer ($wF1 = 0.84386$), and combined ($wF1 = 0.85147$) models. Kerta also produced readability scores with a strong monotonic relationship with the ordinal readability labels (Spearman's $\rho = 0.797$) and demonstrated a greater ability to distinguish readability levels than the Scalabrino ($\rho = 0.5256$) and Buse–Weimer ($\rho = 0.3075$) models. The PLS-SEM analysis further showed that the five dimensions explained 63.5% of the variance in readability, with *Naming Transparency* providing the most dominant contribution.

The Kerta readability score was subsequently integrated into a BM25-based *code search* system and a hybrid BM25+dense vector model using the Java corpus from the *CodeSearchNet Challenge*. The experimental results indicate that readability functions as a secondary signal that improves ranking quality without disrupting the primary ranking structure. Incorporating readability into BM25 produced a consistent performance improvement of +1,99% over the baseline MRR of 0.487. Meanwhile, combining lexical relevance, semantic representations, and readability scores yielded the best-performing hybrid model, with an MRR@10 of 0.533. These findings demonstrate that multidimensionally modeled code readability can be measured effectively and used to improve the quality of *code search* results.

Keywords—Code Readability; Code Search; Conceptual Dimensions of Code Readability; Multi-class Classification; Kerta Metrics; Readability-aware Ranking; Java.