



Penerapan Shannon Entropy untuk Evaluasi Kualitas Kode Program

Inayatullah^{1*}, Nyimas Nisrinaa Kamilah²

¹Prodi Manajemen Informatika, Fakultas Ilmu Komputer dan Rekayasa, Universitas Multi Data Palembang, Palembang

²Prodi Informatika, Fakultas Ilmu Komputer dan Rekayasa, Universitas Multi Data Palembang, Palembang

¹inayatullah@mdp.ac.id, ²nyimasnisrinaakamilah_2226250064@mhs.mdp.ac.id

Abstract

Program code quality is an essential aspect of software development because it affects the system's readability, maintainability, and sustainability. Code quality evaluation is generally carried out using conventional metrics such as Lines of Code and Cyclomatic Complexity. Still, these metrics do not fully capture the regularity of the code structure. This study aims to apply Shannon Entropy as a quantitative indicator in evaluating program code quality. The study uses a quantitative approach with a case study of one software application comprising five source code files. The analysis is carried out through a code tokenization process to obtain the token distribution, followed by the calculation of token probabilities and Shannon Entropy values based on information theory. The results show that the entropy values in each file vary. The file with the highest number of tokens (6,098) produces the highest entropy value of 9.9427, while the file with the lowest number of tokens (435) has the lowest entropy value of 7.3243. Other files show entropy values of 8.1743, 7.6099, and 7.8465. The variation in entropy values reflects the differences in the level of regularity and complexity of the code structure in each file. The research findings show that Shannon Entropy can be used as a complementary metric to evaluate the quality of program code and can quantitatively distinguish the level of regularity in code structure.

Keywords: Entropy Shannon; Kualitas Kode; Penerapan; Program; Tokenisasi;

1. Pendahuluan

Kualitas kode program [1] merupakan salah satu faktor penting dalam pengembangan perangkat lunak yang berkelanjutan [2], [3]. Kode program yang berkualitas tinggi cenderung lebih mudah dipahami, dipelihara, dan dikembangkan serta hemat energi [4], serta memiliki tingkat kesalahan yang lebih rendah. Sebaliknya, kode yang memiliki struktur tidak teratur dan kompleks berpotensi meningkatkan biaya pemeliharaan pada organisasi [5] dan risiko kegagalan sistem [6].

Untuk menilai kualitas kode program, berbagai metrik telah dikembangkan dan digunakan secara luas, seperti Lines of Code (LOC), Cyclomatic Complexity, dan Maintainability Index [7]. Metrik-metrik tersebut memberikan gambaran kuantitatif mengenai ukuran dan kompleksitas kode, namun belum sepenuhnya mampu menggambarkan tingkat keteraturan struktur kode secara menyeluruh. Oleh karena itu, diperlukan pendekatan tambahan yang dapat melengkapi metrik konvensional dalam mengevaluasi kualitas kode program, dalam pengembangan perangkat lunak berkelanjutan [8].

Konsep entropy yang berasal dari teori informasi, khususnya Shannon Entropy, dapat digunakan untuk mengukur tingkat ketidakteraturan atau ketidakpastian suatu sistem [9]. Dalam konteks kode program, entropy dapat merepresentasikan distribusi token kode, seperti keyword, operator, dan identifier. Semakin tidak teratur

distribusi token tersebut, semakin tinggi nilai entropy yang dihasilkan, yang mengindikasikan rendahnya kualitas struktur kode yang dikembangkan [10].

Beberapa penelitian sebelumnya menunjukkan bahwa entropy memiliki potensi sebagai indikator kompleksitas, kualitas perangkat lunak dan pola penyebaran lahan [11]. Namun, penerapan Shannon Entropy sebagai metrik evaluasi kualitas kode program, khususnya dalam konteks studi kasus aplikasi nyata, masih relatif terbatas. Oleh karena itu, penelitian ini bertujuan untuk menerapkan Shannon Entropy dalam mengevaluasi kualitas kode program dan menganalisis hubungannya dengan metrik kualitas kode yang umum digunakan [12].

Hasil penelitian ini diharapkan dapat memberikan kontribusi berupa alternatif metrik kuantitatif yang sederhana namun informatif untuk menilai kualitas kode program yang dikembangkan atau dibangun, khususnya pada pengembangan perangkat lunak skala kecil hingga menengah yang umum dijumpai dalam konteks akademik dan industri lokal. Selain itu, pendekatan Shannon Entropy ini dapat membantu pengembang dalam perbaikan kode program.

Adapun struktur penulisan penelitian ini disusun sebagai berikut. Bagian pertama pendahuluan, bagian kedua menjelaskan metode penelitian yang digunakan,

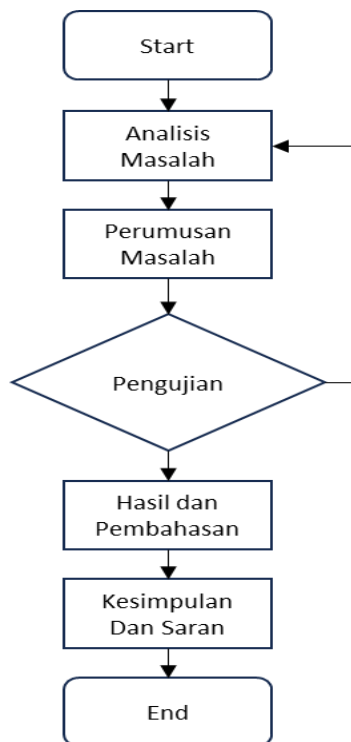
termasuk proses tokenisasi dan perhitungan nilai Shannon Entropy. Bagian ketiga menyajikan hasil penelitian dan pembahasan terkait analisis nilai entropy pada masing-masing file kode. Bagian keempat menyampaikan kesimpulan dan implikasi penelitian, serta peluang pengembangan penelitian selanjutnya.

2. Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode studi kasus. Pendekatan kuantitatif dipilih karena memungkinkan pengukuran kualitas kode program secara objektif dan terukur melalui nilai numerik. Analisis dilakukan terhadap kode sumber dari suatu aplikasi perangkat lunak untuk mengukur nilai Shannon Entropy dan mengevaluasinya sebagai indikator kualitas kode program.

Objek penelitian berupa aplikasi perangkat lunak berbasis bahasa pemrograman tertentu, seperti VB, C# Java, atau PHP yang telah di konversi dalam bentuk txt sehingga memudahkan dalam melakukan analisis. Aplikasi terdiri dari beberapa file kode sumber yang dianalisis untuk memperoleh nilai entropy token dari masing-masing file.

Bagan alir dari penelitian dapat dilihat pada Gambar 1. Dimana terdapat tahapan beberapa tahapan.



Gambar 1. Bagan alir penelitian

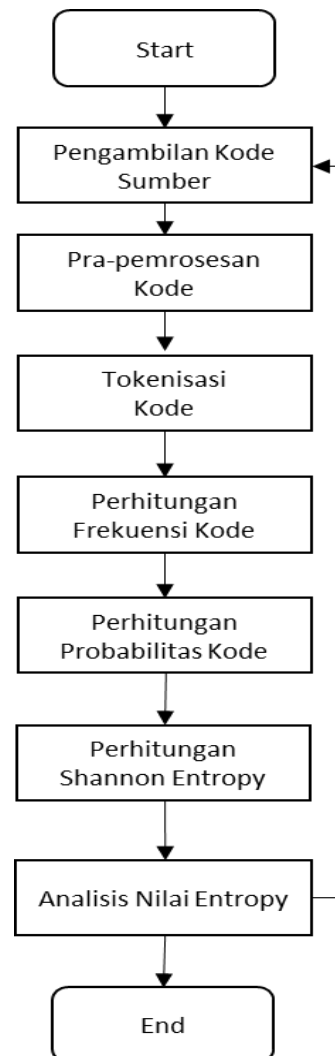
Pengukuran Shannon Entropy dilakukan menggunakan persamaan berikut:

$$H = -\sum(p_i \log_2 p_i) \quad (1)$$

Di mana p_i merupakan probabilitas kemunculan token ke- i terhadap total token dalam kode program. Nilai entropy dihitung untuk setiap kata, kemudian dirata-ratakan untuk memperoleh nilai entropy keseluruhan aplikasi.

Token merupakan unit terkecil yang memiliki makna dalam kode sumber program, seperti keyword, identifier, operator, literal, dan simbol. Token diperoleh melalui proses tokenisasi kode sumber yang bertujuan untuk memecah kode program menjadi elemen-elemen dasar penyusunnya.

Bagan alir dari proses pengujian Entropy dapat dilihat pada Gambar 2.



Gambar 2. Bagan alir proses Shannon entropy

Bagan alir proses Shannon entropy memiliki tahapan sebagai berikut:

1. Pengambilan Kode Sumber
Kode program diambil dari aplikasi yang menjadi objek penelitian.
2. Pra-pemrosesan Kode

Bertujuan menghilangkan elemen yang tidak mempengaruhi struktur kode, seperti komentar dan spasi kosong.

3. Tokenisasi Kode

Kode dipecah menjadi token-token bermakna, meliputi keyword, identifier, operator, literal, dan simbol.

4. Perhitungan Frekuensi Token

Setiap token dihitung jumlah kemunculannya dalam kode program.

5. Perhitungan Probabilitas Token

Probabilitas token dihitung berdasarkan proporsi kemunculannya terhadap total token.

6. Perhitungan Shannon Entropy

Nilai entropy dihitung menggunakan rumus Shannon Entropy untuk mengukur tingkat ketidakaturan kode.

7. Analisis Nilai Entropy

Nilai entropy dianalisis untuk mengevaluasi kualitas dan keteraturan struktur kode program.

Dalam penelitian ini, data yang digunakan memakai dataset tanjak, yang berjumlah 5 file. Potongan kode dari setiap file yang digunakan:

File 1

Module

Global DB As ADODB.Connection

Global rs_Item As ADODB.Recordset

Global rs_Catg As ADODB.Recordset

Global rs_Cari As ADODB.Recordset

Global rs_Sales As ADODB.Recordset

Global rs_Dealer As ADODB.Recordset

Global rs_Cash As ADODB.Recordset

Global rs_Kredit As ADODB.Recordset

Global rs_Sign As ADODB.Recordset

Global SQL As String

Global Data As String

Global DataPublic As String

Global Const Lokasi =

"Provider=SQLOLEDB.1;Persist Security

Info=False;User

ID=sa;Initial Catalog=Olimpia_Motor;Data Source=."

Function Koneksi()

Set DB = New ADODB.Connection

Set rs_Item = New ADODB.Recordset

Set rs_Catg = New ADODB.Recordset

Set rs_Cari = New ADODB.Recordset

File 2

<?php

use Illuminate\Contracts\Http\Kernel;

use Illuminate\Http\Request;

define('LARAVEL_START', microtime(true));

/*

| Check If The Application Is Under Maintenance

| If the application is in maintenance / demo mode via the "down" command

| we will load this file so that any pre-rendered content can be shown

| instead of starting the framework, which could cause an exception.

*/

| Composer provides a convenient, automatically | this application. We just need to utilize it! We'll simply | into the script here so we don't need to manually

File 3

Form Login

Private Sub cmdExit_Click()

End

End Sub

Private Sub cmdLogin_Click()

Form7.Show

End Sub

Private Sub Command1_Click()

frmUbahPassword.Show

End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)

If KeyAscii = 13 Then

Text2.SetFocus

Else

Exit Sub

End If

End Sub

Private Sub Text2_Change()

If keyascii = 13 Then

cmdLogin.SetFocus

End If

End Sub

Form Loading

File 4

'-----

' <auto-generated>

' This code was generated by a tool.

' Runtime Version:2.0.50727.1433

' Changes to this file may cause incorrect behavior and will be lost if

' the code is regenerated.

' </auto-generated>

'-----

Option Strict Off

Option Explicit On

Imports CrystalDecisions.CrystalReports.Engine

Imports CrystalDecisions.ReportSource

Imports CrystalDecisions.Shared

Imports System

Imports System.ComponentModel

Public Class LapBarang

Inherits ReportClass

Public Sub New()

MyBase.New

End Sub

Public Overrides Property ResourceName() As String

Get

Return "LapBarang.rpt"

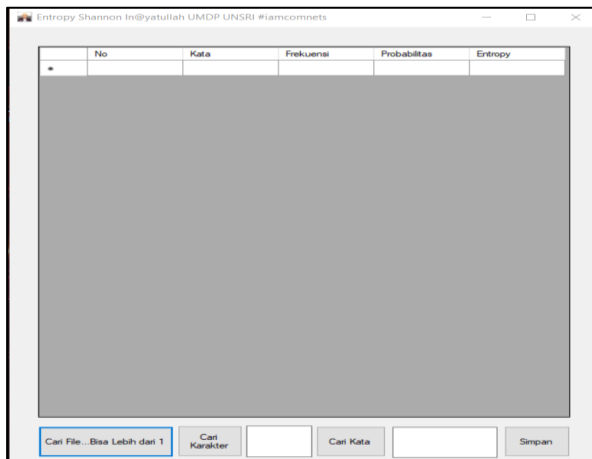
File 5

Public Class Class1

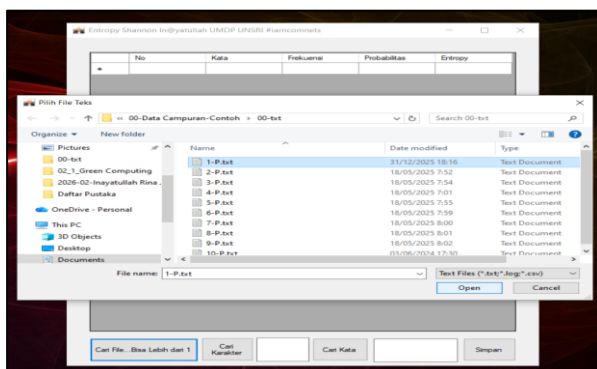
```
Public koneksi As New SqlConnection
Public proses As New SqlCommand
Public cari As SqlCommand
Public database As New SqlDataAdapter
Public tabel As New DataSet
Public Sub buka()
koneksi.ConnectionString = "Data
Source=HERI\SQLEXPRESS;Initial
Catalog=sura;Integrated Security=True"
koneksi.Open()
proses.Connection = koneksi
End Sub
Public Sub tutup()
koneksi.Close()
koneksi.ConnectionString = Nothing
End Sub
End Class
Listing Tampilan Form Login
Public Class Login
Dim cnt As New Class1
Dim login As DataTable
```

3. Hasil dan Pembahasan

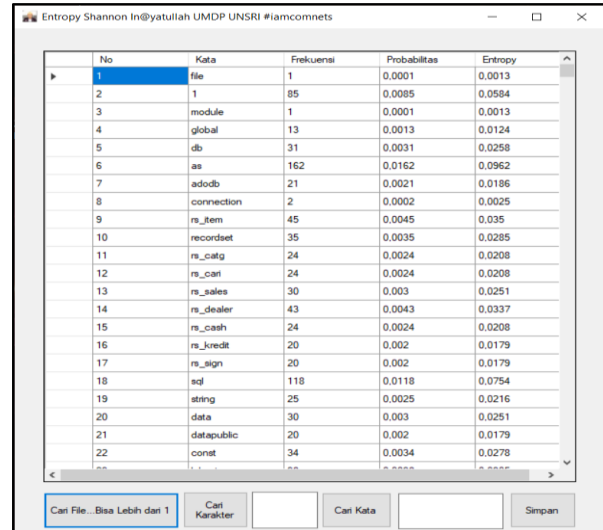
Pada pembahasan dapat diuraikan hasil pengujian dari data yang tersedia, sebanyak 5 file. Pengujian menggunakan aplikasi yang dibangun oleh peneliti.



Gambar 3. Tampilan aplikasi

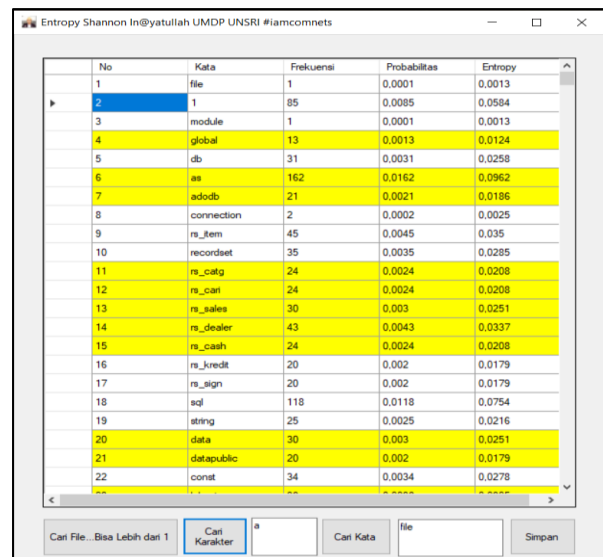


Gambar 4. Tampilan aplikasi cari file



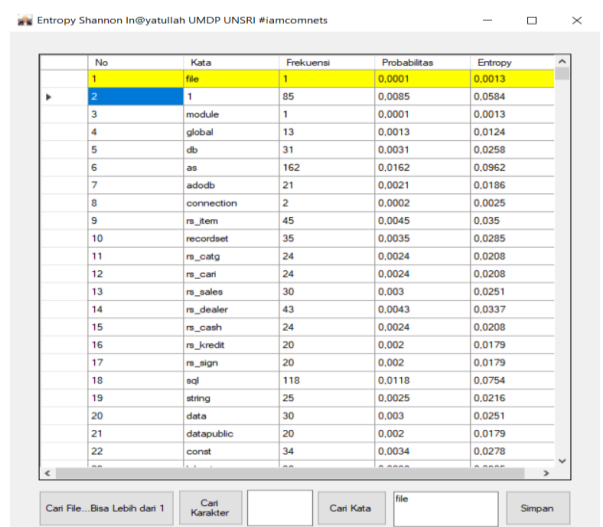
| No | Kata | Frekuensi | Probabilitas | Entropy |
|----|------------|-----------|--------------|---------|
| 1 | file | 1 | 0.0001 | 0.0013 |
| 2 | 1 | 85 | 0.0085 | 0.0584 |
| 3 | module | 1 | 0.0001 | 0.0013 |
| 4 | global | 13 | 0.0013 | 0.0124 |
| 5 | db | 31 | 0.0031 | 0.0258 |
| 6 | as | 162 | 0.0162 | 0.0962 |
| 7 | adodb | 21 | 0.0021 | 0.0186 |
| 8 | connection | 2 | 0.0002 | 0.0025 |
| 9 | rs_item | 45 | 0.0045 | 0.035 |
| 10 | recordset | 35 | 0.0035 | 0.0285 |
| 11 | rs_catg | 24 | 0.0024 | 0.0208 |
| 12 | rs_cari | 24 | 0.0024 | 0.0208 |
| 13 | rs_sales | 30 | 0.003 | 0.0251 |
| 14 | rs_dealer | 43 | 0.0043 | 0.0337 |
| 15 | rs_cash | 24 | 0.0024 | 0.0208 |
| 16 | rs_kredit | 20 | 0.002 | 0.0179 |
| 17 | rs_sign | 20 | 0.002 | 0.0179 |
| 18 | sql | 118 | 0.0118 | 0.0754 |
| 19 | string | 25 | 0.0025 | 0.0216 |
| 20 | data | 30 | 0.003 | 0.0251 |
| 21 | datapublic | 20 | 0.002 | 0.0179 |
| 22 | const | 34 | 0.0034 | 0.0278 |

Gambar 5. Tampilan aplikasi hasil proses file



| No | Kata | Frekuensi | Probabilitas | Entropy |
|----|------------|-----------|--------------|---------|
| 1 | file | 1 | 0.0001 | 0.0013 |
| 2 | 1 | 85 | 0.0085 | 0.0584 |
| 3 | module | 1 | 0.0001 | 0.0013 |
| 4 | global | 13 | 0.0013 | 0.0124 |
| 5 | db | 31 | 0.0031 | 0.0258 |
| 6 | as | 162 | 0.0162 | 0.0962 |
| 7 | adodb | 21 | 0.0021 | 0.0186 |
| 8 | connection | 2 | 0.0002 | 0.0025 |
| 9 | rs_item | 45 | 0.0045 | 0.035 |
| 10 | recordset | 35 | 0.0035 | 0.0285 |
| 11 | rs_catg | 24 | 0.0024 | 0.0208 |
| 12 | rs_cari | 24 | 0.0024 | 0.0208 |
| 13 | rs_sales | 30 | 0.003 | 0.0251 |
| 14 | rs_dealer | 43 | 0.0043 | 0.0337 |
| 15 | rs_cash | 24 | 0.0024 | 0.0208 |
| 16 | rs_kredit | 20 | 0.002 | 0.0179 |
| 17 | rs_sign | 20 | 0.002 | 0.0179 |
| 18 | sql | 118 | 0.0118 | 0.0754 |
| 19 | string | 25 | 0.0025 | 0.0216 |
| 20 | data | 30 | 0.003 | 0.0251 |
| 21 | datapublic | 20 | 0.002 | 0.0179 |
| 22 | const | 34 | 0.0034 | 0.0278 |

Gambar 6. Tampilan aplikasi cari karakter



| No | Kata | Frekuensi | Probabilitas | Entropy |
|----|------------|-----------|--------------|---------|
| 1 | file | 1 | 0.0001 | 0.0013 |
| 2 | 1 | 85 | 0.0085 | 0.0584 |
| 3 | module | 1 | 0.0001 | 0.0013 |
| 4 | global | 13 | 0.0013 | 0.0124 |
| 5 | db | 31 | 0.0031 | 0.0258 |
| 6 | as | 162 | 0.0162 | 0.0962 |
| 7 | adodb | 21 | 0.0021 | 0.0186 |
| 8 | connection | 2 | 0.0002 | 0.0025 |
| 9 | rs_item | 45 | 0.0045 | 0.035 |
| 10 | recordset | 35 | 0.0035 | 0.0285 |
| 11 | rs_catg | 24 | 0.0024 | 0.0208 |
| 12 | rs_cari | 24 | 0.0024 | 0.0208 |
| 13 | rs_sales | 30 | 0.003 | 0.0251 |
| 14 | rs_dealer | 43 | 0.0043 | 0.0337 |
| 15 | rs_cash | 24 | 0.0024 | 0.0208 |
| 16 | rs_kredit | 20 | 0.002 | 0.0179 |
| 17 | rs_sign | 20 | 0.002 | 0.0179 |
| 18 | sql | 118 | 0.0118 | 0.0754 |
| 19 | string | 25 | 0.0025 | 0.0216 |
| 20 | data | 30 | 0.003 | 0.0251 |
| 21 | datapublic | 20 | 0.002 | 0.0179 |
| 22 | const | 34 | 0.0034 | 0.0278 |

Gambar 7. Tampilan aplikasi cari kata

Tabel 1. Hasil pengujian file 1

| No | Token | Frekuensi | Probabilitas | Entropy |
|-------|-------------|-----------|--------------|---------|
| 1 | file | 1 | 0,0001 | 0,0013 |
| 2 | 1 | 85 | 0,0085 | 0,0584 |
| 3 | module | 1 | 0,0001 | 0,0013 |
| 4 | global | 13 | 0,0013 | 0,0124 |
| 5 | db | 31 | 0,0031 | 0,0258 |
| 6 | as | 162 | 0,0162 | 0,0962 |
| 7 | adodb | 21 | 0,0021 | 0,0186 |
| 8 | connection | 2 | 0,0002 | 0,0025 |
| 9 | rs_item | 45 | 0,0045 | 0,035 |
| 10 | recordset | 35 | 0,0035 | 0,0285 |
| | | | | |
| 943 | datetime | 7 | 0,0007 | 0,0073 |
| 944 | 25 | 4 | 0,0004 | 0,0045 |
| 945 | 40 | 1 | 0,0001 | 0,0013 |
| 946 | 15 | 2 | 0,0002 | 0,0025 |
| 947 | statusbayar | 2 | 0,0002 | 0,0025 |
| File | 947 | 10024 | 1 | 8,1743 |

Ke:1

File ke-1 memiliki jumlah token sebanyak 947 dengan jumlah frekuensi token 10.024 dan menghasilkan nilai entropy sebesar 8,1743. Nilai entropy yang relatif tinggi ini mengindikasikan bahwa struktur kode pada file tersebut memiliki tingkat ketidakteraturan yang cukup besar, yang kemungkinan disebabkan oleh penggunaan berbagai jenis token dengan distribusi yang beragam. Hal ini dapat berkaitan dengan kompleksitas logika atau panjang fungsi dalam file tersebut.

Tabel 2. Hasil Pengujian file 2

| No | Token | Frekuensi | Probabilitas | Entropy |
|-------|----------------|-----------|--------------|---------|
| 1 | file | 37 | 0,001 | 0,0102 |
| 2 | 2 | 209 | 0,0058 | 0,0432 |
| 3 | php | 100 | 0,0028 | 0,0236 |
| 4 | use | 209 | 0,0058 | 0,0432 |
| 5 | illuminate | 241 | 0,0067 | 0,0485 |
| 6 | contracts | 8 | 0,0002 | 0,0027 |
| 7 | http | 215 | 0,006 | 0,0442 |
| 8 | kernel | 16 | 0,0004 | 0,005 |
| 9 | request | 234 | 0,0065 | 0,0473 |
| 10 | define | 10 | 0,0003 | 0,0033 |
| | | | | |
| 6094 | constraint | 8 | 0,0002 | 0,0027 |
| 6095 | foreign | 8 | 0,0002 | 0,0027 |
| 6096 | references | 8 | 0,0002 | 0,0027 |
| 6097 | pegawai_ibfk_1 | 1 | 0 | 0,0004 |
| 6098 | commit | 1 | 0 | 0,0004 |
| File | 6098 | 35902 | 1 | 9,9427 |

Ke:2

File ke-2 menunjukkan jumlah token paling besar, yaitu 6.098 token dengan frekuensi token sebesar 35.902, serta menghasilkan nilai entropy tertinggi di antara seluruh file, yaitu 9,9427. Tingginya nilai entropy pada file ini mengindikasikan tingkat kompleksitas dan variasi struktur kode yang paling tinggi dibandingkan file lainnya. Hal ini menunjukkan bahwa file ke-2 berpotensi memiliki struktur kode yang lebih kompleks dan kurang terstruktur dibandingkan file lain dalam aplikasi yang sama.

Tabel 3. Hasil Pengujian file 3

| No | Token | Frekuensi | Probabilitas | Entropy |
|----|-------|-----------|--------------|---------|
| 1 | file | 1 | 0,0001 | 0,0016 |
| 2 | 3 | 2 | 0,0003 | 0,003 |
| 3 | form | 20 | 0,0025 | 0,0219 |

| | | | | |
|-------|----------------|-------|--------|--------|
| 4 | login | 2 | 0,0003 | 0,003 |
| 5 | private | 121 | 0,0154 | 0,0926 |
| 6 | sub | 307 | 0,039 | 0,1825 |
| 7 | cmdexit_click | 1 | 0,0001 | 0,0016 |
| 8 | end | 247 | 0,0314 | 0,1567 |
| 9 | cmdlogin_click | 1 | 0,0001 | 0,0016 |
| 10 | form7 | 1 | 0,0001 | 0,0016 |
| | | | | |
| 594 | viewambildarah | 1 | 0,0001 | 0,0016 |
| 595 | expr2 | 1 | 0,0001 | 0,0016 |
| 596 | inner | 3 | 0,0004 | 0,0043 |
| 597 | join | 3 | 0,0004 | 0,0043 |
| 598 | off | 1 | 0,0001 | 0,0016 |
| File | 598 | 7875 | 1 | 7,6099 |

Ke:3

File ke-3 memiliki jumlah token sebanyak 598 dengan frekuensi token 7.875 dan nilai entropy sebesar 7,6099. Nilai entropy yang lebih rendah dibandingkan file ke-1 dan ke-2 menunjukkan bahwa struktur kode pada file ini relatif lebih teratur. Distribusi token yang lebih terkendali mengindikasikan bahwa kode pada file ini cenderung lebih sederhana dan mudah dipahami.

Tabel 4. Hasil Pengujian file 4

| No | Token | Frekuensi | Probabilitas | Entropy |
|-------|--------------|-----------|--------------|---------|
| 1 | file | 7 | 0,0004 | 0,0047 |
| 2 | 4 | 14 | 0,0008 | 0,0086 |
| 3 | auto | 4 | 0,0002 | 0,0029 |
| 4 | generated | 8 | 0,0005 | 0,0053 |
| 5 | this | 13 | 0,0008 | 0,008 |
| 6 | code | 17 | 0,001 | 0,0101 |
| 7 | was | 2 | 0,0001 | 0,0016 |
| 8 | by | 28 | 0,0017 | 0,0155 |
| 9 | a | 2 | 0,0001 | 0,0016 |
| 10 | tool | 2 | 0,0001 | 0,0016 |
| | | | | |
| 1149 | 06 | 1 | 0,0001 | 0,0008 |
| 1150 | pk_pelanggan | 1 | 0,0001 | 0,0008 |
| 1151 | pk_tender | 1 | 0,0001 | 0,0008 |
| File | 1151 | 16687 | 1 | 7,8465 |

Ke:4

File ke-4 memiliki jumlah token 1.151 dengan frekuensi token 16.687 dan menghasilkan nilai entropy sebesar 7,8465. Nilai ini menunjukkan tingkat ketidakteraturan yang sedang, di mana struktur kode tidak terlalu kompleks namun masih memiliki variasi token yang cukup signifikan. Hal ini dapat disebabkan oleh adanya beberapa logika percabangan atau pemanggilan fungsi yang cukup intensif pada aplikasi yang dikembangkan atau dibangun.

Tabel 5. Hasil Pengujian file 5

| No | Token | Frekuensi | Probabilitas | Entropy |
|-------|---------------|-----------|--------------|---------|
| 1 | file | 1 | 0,0001 | 0,0017 |
| 2 | 5 | 9 | 0,0012 | 0,0118 |
| 3 | public | 22 | 0,003 | 0,025 |
| 4 | class | 30 | 0,0041 | 0,0323 |
| 5 | class1 | 10 | 0,0014 | 0,0129 |
| 6 | koneksi | 95 | 0,0129 | 0,0809 |
| 7 | as | 305 | 0,0414 | 0,1901 |
| 8 | new | 133 | 0,018 | 0,1045 |
| 9 | sqlclient | 72 | 0,0098 | 0,0652 |
| 10 | sqlconnection | 31 | 0,0042 | 0,0332 |
| | | | | |
| 431 | p | 5 | 0,0007 | 0,0071 |
| 432 | b | 4 | 0,0005 | 0,0059 |

| | | | | |
|-----------|-------|------|--------|--------|
| 433 | inner | 1 | 0,0001 | 0,0017 |
| 434 | join | 1 | 0,0001 | 0,0017 |
| 435 | on | 1 | 0,0001 | 0,0017 |
| File Ke:5 | 435 | 7372 | 1 | 7,3243 |

File ke-5 memiliki jumlah token paling sedikit, yaitu 435 token dengan frekuensi token 7.372 dan nilai entropy terendah sebesar 7,3243. Rendahnya nilai entropy pada file ini mengindikasikan bahwa struktur kode relatif lebih sederhana dan terorganisasi dengan baik. File ini berpotensi memiliki kualitas kode yang lebih baik dibandingkan file lainnya berdasarkan indikator Shannon Entropy.

Tabel 6. Komparasi dari 5 file

| File Ke | Jumlah Token | Jumlah Frekuensi | Total Probabilitas Token | Jumlah Entropy |
|---------|--------------|------------------|--------------------------|----------------|
| 1 | 947 | 10024 | 1 | 8,1743 |
| 2 | 6098 | 35902 | 1 | 9,9427 |
| 3 | 598 | 7875 | 1 | 7,6099 |
| 4 | 1151 | 16687 | 1 | 7,8465 |
| 5 | 435 | 7372 | 1 | 7,3243 |

Berdasarkan Tabel 6, terlihat bahwa kelima file kode program memiliki karakteristik yang berbeda-beda dari sisi jumlah token, frekuensi token, dan nilai Shannon Entropy. Perbedaan karakteristik tersebut menunjukkan adanya variasi tingkat keteraturan dan kompleksitas struktur kode pada masing-masing file yang dianalisis.

Setiap file kode merepresentasikan pola distribusi token yang berbeda sesuai dengan fungsi dan struktur logika yang diimplementasikan. Variasi distribusi token ini secara langsung memengaruhi nilai Shannon Entropy yang dihasilkan, sehingga mencerminkan perbedaan tingkat ketidakteraturan struktur kode antar file.

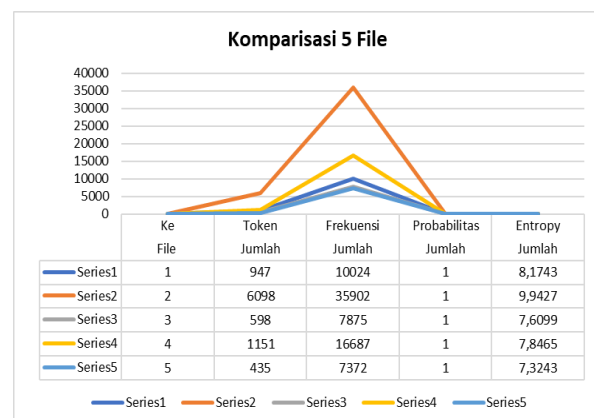
File dengan jumlah token yang lebih besar cenderung memiliki variasi token yang lebih beragam, yang berdampak pada meningkatnya nilai Shannon Entropy. Kondisi ini mengindikasikan bahwa struktur kode pada file tersebut relatif lebih kompleks dan memiliki tingkat ketidakteraturan yang lebih tinggi, yang berpotensi memengaruhi kemudahan pemahaman dan pemeliharaan kode.

Sebaliknya, file dengan jumlah token yang lebih sedikit menunjukkan distribusi token yang lebih terkonsentrasi. Distribusi token yang lebih terpusat ini menghasilkan nilai entropy yang lebih rendah, yang mengindikasikan struktur kode yang lebih sederhana, lebih konsisten, dan relatif lebih mudah dipahami serta dipelihara.

Selain jumlah token, perbedaan frekuensi kemunculan token juga berpengaruh terhadap nilai Shannon Entropy. File yang memiliki frekuensi token yang lebih merata cenderung menghasilkan nilai entropy yang lebih tinggi dibandingkan dengan file yang didominasi oleh beberapa jenis token tertentu. Distribusi token yang merata mencerminkan variasi struktur dan logika program yang lebih kompleks.

Dengan demikian, hasil komparasi pada Tabel 6 menunjukkan bahwa Shannon Entropy mampu membedakan tingkat keteraturan struktur kode antar file secara kuantitatif. Nilai entropy yang dihasilkan memberikan gambaran tambahan mengenai kualitas struktur kode program yang tidak sepenuhnya dapat direpresentasikan oleh metrik konvensional seperti *Lines of Code*. Temuan ini memperkuat peran Shannon Entropy sebagai metrik pelengkap dalam evaluasi kualitas kode program.

Pada Gambar 3, dapat dilihat grafik hasil dari komparasi 5 file yang dilakukan pengujian. Secara keseluruhan, Grafik komparasi ini memperkuat temuan bahwa Shannon Entropy mampu merepresentasikan perbedaan tingkat keteraturan dan kompleksitas struktur kode program secara kuantitatif. Perbedaan nilai entropy antar file memberikan gambaran tambahan mengenai kualitas struktur kode yang tidak sepenuhnya dapat ditangkap oleh metrik konvensional seperti *Lines of Code*, sehingga memperkuat peran Shannon Entropy sebagai metrik pelengkap dalam evaluasi kualitas kode program.



Gambar 8. Komparasi 5 File

4. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa Shannon Entropy dapat diterapkan sebagai indikator untuk mengevaluasi kualitas kode program. Nilai entropy yang tinggi menunjukkan tingkat ketidakteraturan struktur kode yang lebih besar, sedangkan nilai entropy yang rendah mengindikasikan kode yang lebih terstruktur dan berkualitas.

Penerapan Shannon Entropy dapat menjadi metrik pelengkap bagi metrik kualitas kode konvensional dalam menilai kompleksitas dan keteraturan kode program. Penelitian ini diharapkan dapat menjadi referensi awal bagi pengembang dan peneliti dalam mengkaji kualitas kode program menggunakan pendekatan berbasis teori informasi.

Hasil dari pengukuran menunjukkan bahwa nilai Shannon Entropy pada kelima file kode sumber

memiliki variasi yang berbeda-beda. Perbedaan nilai entropy tersebut mencerminkan tingkat keteraturan struktur kode yang tidak sama pada setiap file. File dengan struktur kode yang lebih kompleks dan banyak percabangan logika cenderung memiliki nilai entropy yang lebih tinggi dibandingkan file dengan struktur yang lebih sederhana.

Referensi

- [1] T. Nazaré, J. Gadelha, E. Nepomuceno, and R. Lozi, "Green Computing for Energy Transition: A Survey," *IEEE Lat. Am. Trans.*, vol. 21, no. 9, pp. 937–948, 2023, doi: 10.1109/tla.2023.10251799.
- [2] G. Ruhe, "Optimization in Software Engineering: A Pragmatic Approach," *Contemp. Empir. Methods Softw. Eng.*, pp. 235–261, 2020, doi: 10.1007/978-3-030-32489-6_9.
- [3] P. P. Dwivedi and D. K. Sharma, "Application of Shannon Entropy and COCOSO techniques to analyze performance of sustainable development goals: The case of the Indian Union Territories," *Results Eng.*, vol. 14, p. 100416, 2022, doi: <https://doi.org/10.1016/j.rineng.2022.100416>.
- [4] J. Penerapan, T. Informasi, D. Suatu, and A. Berbasis, "ANALISIS QUALITY CODE MENGGUNAKAN SONARQUBE DALAM SUATU APLIKASI BERBASIS LARAVEL," vol. 02, pp. 89–103, 2023.
- [5] R. Chourasiya, "Impact of Green Computing Practices in Organization," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 6, no. 1, pp. 1046–1053, 2021, doi: 10.48175/568.
- [6] R. Sehgal, D. Mehrotra, R. Nagpal, and R. Sharma, "Green software: Refactoring approach," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 7, pp. 4635–4643, 2022, doi: 10.1016/j.jksuci.2020.10.022.
- [7] M. Andryan, W. Saputra, W. A. Riuditama, and H. Setyowati, "Survei Teknik-Teknik Pengukuran Kualitas Perangkat Lunak," vol. 3, no. 1, pp. 2715–2731, 2021.
- [8] S. Nazir, N. Fatima, S. Chuprat, H. Sarkan, F. Nurulhuda, and N. N. A. Sjarif, "Sustainable Software Engineering : A Perspective of Individual Sustainability," vol. 10, no. 2, pp. 676–683, 2020.
- [9] Z. Zheng, "Shannon Theory BT - Modern Cryptography Volume 1: A Classical Introduction to Informational and Mathematical Principle," Z. Zheng, Ed. Singapore: Springer Singapore, 2022, pp. 91–151.
- [10] S. Naeem, S. Anam, A. Ali, M. Zubair, and M. M. Ahmed, "A Brief History of Information Theory by Claude Shannon in Data Communication," no. 13.
- [11] B. Gizela, D. Puspita, M. Inri, and S. Pamungkas, "Pengaruh Perkembangan Industri Terhadap Pola Penyebaran Lahan Terbangun di Kota Balikpapan Berbasis Analisis Shannon ' s Entropy," vol. 05, no. 01, pp. 26–32, 2025.
- [12] H. Farasat, M. Sayadi, and H. Y. Gachini, "Analysis of the Contents of the Science Textbooks of the First Elementary Period , based on the Amount Of Attention to Science Process Skills with Shannon Entropy Method هجوت نازيم ساسا رب ييادتبا لوا قروء مولع ياه باتك ياتوحم ليبلحت نوناش بيورتتا شور باب يملع دنيارف ياه تراهم هب," vol. 12, no. 1, pp. 667–689, 2023.