

**OPTIMASI *MAINTAINABILITY* APLIKASI DENGAN
*PRINSIP CLEAN CODE***

SKRIPSI

Diajukan untuk Memenuhi Tugas Akhir dan Melengkapi Syarat
Guna Memperoleh Gelar Sarjana Strata Satu (S-1) dalam
Teknologi Informasi



**Diajukan Oleh :
Siti Zikrina Farahdiba
NIM : 2108096004**

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO
SEMARANG
2025**

PERNYATAAN KEASLIAN

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Siti Zikrina Farahdiba

NIM : 2108096004

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul:

OPTIMASI *MAINTAINABILITY* APLIKASI DENGAN PRINSIP *CLEAN CODE*

Secara keseluruhan adalah hasil penelitian/karya saya sendiri, kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 24 April 2025

Pembuat Pernyataan



Siti Zikrina Farahdiba

NIM. 2108096004

PENGESAHAN



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Hamka Ngaliyan Semarang
Telp.024-7601295 Fax.7615387

PENGESAHAN

Naskah proposal skripsi berikut ini:

Judul : Optimasi *Maintainability* Aplikasi Dengan Prinsip *Clean Code*

Penulis : Siti Zikrina Farahdiba

NIM : 2108096004

Jurusan : Teknologi Informasi

Telah diujikan dalam seminar sidang akhir oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana bidang ilmu Teknologi Informasi.

Semarang, 22 April 2025

DEWAN PENGUJI

Penguji I

Hery Mastofa, M.Kom.
NIP. 198703172019031000

Penguji II

Nur Cahyo Hendro Wibowo, S.T., M.Kom.
NIP. 197312222006041001

Penguji III

Dr. Masy Ari Ulinuha, ST., M.T
NIP. 198108122011011007

Penguji IV

Adhwal Arwani Mahfudh, M.Kom.
NIP. 199107032019031006

Pembimbing I

Nur Cahyo Hendro Wibowo, S.T., M.Kom.
NIP. 197312222006041001

Pembimbing II

Siti Nur'Aini, M.Kom.
NIP.198401312018012001

NOTA DINAS

NOTA DINAS

Semarang , 18 Maret 2025

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. Wb. Wb.

Dengan ini diberitahukan bahwa saya telah melakukakn bimbingan,
arahan dan koreksi
naskah skripsi dengan :

Judul : OPTIMASI MAINTAINABILITY APLIKASI DENGAN
PRINSIP CLEAN CODE (STUDI KASUS: WEBSITE
MONITORING SISTEM JURNAL)
Nama : Siti Zikrina Farahdiba
NIM : 2108096004
Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah diajukan
kepada Fakultas Sains dan Teknologi UIN Walisongo Semarang
untuk diuji dalam Sidang Munaqsyah.
Wassalamu'alaikum. Wb. Wb.

Pembimbing 1,



Nur Cahyo Hendro Wibowo, S.T., M.Kom
NIP.197312222006041001

NOTA DINAS

NOTA DINAS

Semarang , 17 Maret 2025

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. Wb. Wb.

Dengan ini diberitahukan bahwa saya telah melakukakn bimbingan,
arahan dan koreksi
naskah skripsi dengan :

Judul : OPTIMASI MAINTAINABILITY APLIKASI DENGAN
PRINSIP CLEAN CODE (STUDI KASUS: WEBSITE
MONITORING SISTEM JURNAL)
Nama : Siti Zikrina Farahdiba
NIM : 2108096004
Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah diajukan
kepada Fakultas Sains dan Teknologi UIN Walisongo Semarang
untuk diuji dalam Sidang Munaqsyah.
Wassalamu'alaikum. Wb. Wb.

Pembimbing 2,



Siti Nur'aini M.Kom.
NIP.198401312018012001

HALAMAN PERSEMBAHAN

Puji dan syukur penulis panjatkan kepada Allah SWT , dengan rahmat-Nya , laporan tugas akhir ini dapat diselesaikan dengan baik. Sebagai bentuk penghormatan dan rasa terima kasih, karya ini penulis persembahkan kepada:

1. Kedua orang tua tercinta yang selalu mendoakan dan mendukung tanpa henti.
2. Kedua bibi dan Kakak tersayang yang selalu memberikan kasih sayang dan mendoakan pada setiap langkah perjalanan ini
3. Dosen-Dosen Program Studi Teknologi Informasi yang telah membimbing, mengarahkan dan memberikan ilmu yang sangat berharga selama masa perkuliahan .
4. Sahabat Terbaik, Ananda Khoirun nisa' dan Bellin Nabilah yang selalu menemani dan mendukung penulis.
5. Terakhir dan yang paling penting , saya ingin berterima kasih kepada diri saya sendiri karena telah percaya, bekerja keras, tidak menyerah dan selalu berusaha memberikan yang terbaik.

MOTTO

مَا وَدَّعَكَ رَبُّكَ وَمَا قَلَىٰ ﴿٣﴾

“Tuhanmu , tidak meninggalkanmu dan tidak membencimu”

— Ad-Dhuha: 3 —

“The more you know , the more you don’t know”

— Aristotle —

ABSTRAK

Maintainability merupakan faktor penting dalam pengembangan perangkat lunak untuk memastikan kode dapat dipelihara dan diperbaiki. kompleksitas yang tinggi dapat menurunkan *maintainability*, sehingga menyulitkan proses pengembangan dan perawatan di masa depan, Penelitian ini menerapkan prinsip *clean code* pada Website Monitoring Sistem Jurnal guna meningkatkan *Maintainability Index*

Metode penelitian yang digunakan adalah metode kuantitatif komperatid mean dengan membandingkan *Maintainability Indeks* sebelum dan sesudah penerapan perinsip *clean code*. Pengukuran dilakukan menggunakan metrik seperti *Lines of code (LOC)*, *Cyclomatic Complexity (CC)* dan *Helstead Volume (HV)*, dengan *SonarQube* sebagai alat deteksi *code smell*.

Hasil penelitian menunjukkan bahwa penerapan prinsip *Clean Code* secara signifikan meningkatkan *Maintainability Indeks*, dari rata-rata sebesar 54.57 sebelum *refactoring* menjadi 76.43 setelah *refactoring*, atau mengalami peningkatan sebesar sebesar 40,06%, mengurangi kompleksitas, serta meningkatkan keterbacaan dan pemeliharaan kode.

Kata kunci : *Clean Code*, *Maintainability Indeks*, *Refactoring*, *Code Smell*

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir ini yang berjudul “**Optimasi *Maintainability* Aplikasi dengan Prinsip *Clean Code***” dengan baik. Tugas Akhir ini Disusun Sebagai syarat untuk memperoleh gelar sarjana pada Program Studi Teknologi Informasi, UIN Walisongo Semarang.

Dalam penyusunan Tugas Akhir ini, penulis menerima banyak dukungan dan bimbingan dari berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih yang sebesar besarnya kepada:

1. Ibu Misniati dan Bapak Miftahul Ilmi ,selaku kedua orang tua penulis, yang selalu memberikan kasih sayang, doa, serta dukungan tanpa henti.
2. Ibu sofia dan Ibu salmiah, selaku Bibi penulis, atas perhatian, kepedulian dan motivasi yang senantiasa diberikan selama perjalanan akademik ini.
3. Bapak Dr. Khotibul Umam, S.T., M.Kom., selaku Ketua program Studi Teknologi Informasi UIN Walisongo Semarang, atas bimbingan dan dukungan selama masa studi
4. Bapak Nur Cahyo Hendro Wibowo, S.T., M.Kom., selaku Dosen Pembimbing I sekaligus Dosen Wali , atas

- bimbingan, saran dan wawasan berharga yang membantu dalam penyelesaian penelitian ini.
5. Ibu Siti Nur'aini, M.Kom., selaku Dosen Pembimbing II, atas bimbingan, saran , dan wawasan berharga yang membantu dalam penyelesaian penelitian ini.
 6. Seluruh Dosen Program Studi Teknologi Informasi serta staf akademik UIN Walisongo Semarang, atas ilmu, pengalaman dan bantuan yang telah diberikan selama masa perkuliahan
 7. Seluruh pihak yang telah membantu penyusunan skripsi ini, yang tidak dapat disebutkan satu per satu. Terima Kasih atas segala bentuk dukungan dan bantuanya.

Penulis berharap skripsi ini dapat bermanfaat untuk seluruh pihak dibidang teknologi Informasi terutama bagi civitas akademika UIN Walisongo Semarang. Segala masukan dan saran yang membangun sangat diharapkan untuk penyempurnaan di masa mendatang

Semarang, 15 Maret
2025

Penulis

DAFTAR ISI

PERNYATAAN KEASLIAN	i
PENGESAHAN	ii
NOTA DINAS.....	iii
NOTA DINAS.....	iv
HALAMAN PERSEMBAHAN	v
MOTTO.....	vi
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
BAB I.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah	8
1.3 Rumusan Masalah	8
1.4 Batasan Masalah	9
1.5 Tujuan Penelitian	9
1.6 Manfaat Penelitian	10
1.7 Sistematika Penulisan.....	11
BAB II.....	14
2.1 Kajian Pustaka	14
2.2 Landasan Teori.....	43
2.3 Perbandingan Penelitian	51
BAB III METODE PENELITIAN	56
3.1 Alur Kerja Penelitian.....	57

3.2	Subjek Penelitian.....	61
3.3	Kebutuhan Perangkat Penelitian.....	61
BAB IV		64
4.1	Pengumpulan Data Smell Code.....	64
4.2	Pengukuran Maintainability Indeks Sebelum Refactorisasi	65
4.3	Melakukan Refactoring (penerapan Prinsip <i>Clean Code</i>)	84
4.4	Pengukuran Nilai <i>Maintainability Index</i> sesudah diterapkannya <i>clean code</i>	115
4.5	Analisis Temuan dan Hasil Penelitian.	132
BAB V		141
5.1	Saran.....	141
5.2	Kesimpulan.....	141
DAFTAR PUSTAKA		144

DAFTAR GAMBAR

gambar 3.1 Alur kerja Penelitian	57
gambar 4. 1 Hasil Perhitungan LOC file Py Sebelum Refactoring.....	67
gambar 4. 2 Hasil Perhitungan LOC PHP Sebelum Refactoring.....	69
gambar 4. 3 Hasil Perhitungan CC File Py Sebelum Refactoring.....	72
gambar 4. 4 Hasil Perhitungan CC file PHP Sebelum Refactoring.....	74
gambar 4. 13 Hasil HV file PHP Sesudah Refactoring	129
gambar 4. 14 Perhitungan MI Sesudah Refactoring	131
gambar 4. 15 Hasil Uji-t	136

DAFTAR TABEL

Table 2. 1 Klasifikasi Maintainability Index.....	17
Tabel 2.2 Perbandingan Penelitian.....	43
Tabel 3.1 Kebutuhan Perangkat Keras.....	52
Tabel 3.2 Kebutuhan Perangkat Lunak.....	63
Tabel 4. 1 LOC Sebelum Refactoring.....	70
Tabel 4. 2 CC Sebelum Refactoring.....	75
Tabel 4. 3 HV Sebelum Refactoring	81
Tabel 4. 4 MI Sebelum Refactoring.....	83
Tabel 4. 5 LOC Sesudah Refactoring.....	120
Tabel 4. 6 CC Sesudah Refactoring.....	125
Tabel 4. 7 HV Setelah Refactoring.....	129
Tabel 4. 8 MI Sesudah Refactoring.....	133
Tabel 4. 9 Perbandingan MI.....	132
Tabel 4. 10 Hasil Uji Mean.....	133

DAFTAR LAMPIRAN

Lampiran 1 Skrip Menghitung nilai LOC,CC,dan HV untuk file Py	149
Lampiran 2 Skrip Menghitung nilai LOC,CC,dan HV untuk file PHP	151
Lampiran 3 Skrip Kode halaman_Artikel.php sebelum penerapan prinsip Clean Code	154
Lampiran 4 Skrip Kode Halaman_Artikel.phph Sesudah penerapan Prinsip Clean Code	175
Lampiran 5 Skrip Kode indeks.php Sebelum penerapan Prinsiap Clean Code	184
Lampiran 6 Skrip indeks.php Sesudah Penerapan Prinsip Clean Code	190
Lampiran 7 Skrip kode Halaman_Jurnal.php Sebelum penerapan Prinsip Clean Code	192
Lampiran 8 Skrip kode Halaman_Jurnal.php Sesudah penerapan Prinsip Clean Code	202
Lampiran 9 Skrip kode Halaman_Home.php Sebelum penerapan Prinsip Clean Code	209
Lampiran 10 Skrip Kode Halaman_Home.php sesidah penerapan prinsip Clean Code	215
Lampiran 11 Skrip Kode IndexScraper.py Sebelum penerapan Prinsip Clean Code	218

Lampiran 12 Skrip Kode IndexScraper.py sesudah penerapan	
Prinsip Clean Code	222
Lampiran 13 Skrip JurnalScraper.py Sebelum menerapkan	
Prinsip Clean Code	225
Lampiran 14 Skrip JurnalScraper.py Sesudah penerapan	
Prinsip Clean Code	236

BAB I

PENDAHULUAN

1.1 Latar Belakang

Clean code adalah kode yang mudah dibaca, dimengerti, dan dikelola oleh pengembang lain. Kode bersih juga mengikuti prinsip-prinsip desain yang baik dan memiliki struktur yang terorganisir dengan baik (Wijaya & Karmilasari 2021). Penerapan *clean code* yang mengacu pada buku "*Clean code : A Handbook of Agile Software Craftsmanship*" karya Robert C. Martin, sangat penting karena membantu menghindari masalah yang sering terjadi dalam pengembangan perangkat lunak, seperti *code smell* dengan indikator adanya masalah dalam kode yang dapat memperumit proses pemeliharaan. Dengan menerapkan *clean code*, kode menjadi lebih terstruktur, lebih mudah dipahami dan lebih mudah dimodifikasi tanpa menimbulkan masalah baru.

Setelah prinsip kode diterapkan, Pengukuran *maintainability indeks* menjadi relevan untuk memastikan bahwa kode tersebut mudah untuk dipelihara dalam jangka panjang. *Maintainability* pada perangkat lunak merujuk pada kemampuan kode untuk dimodifikasi dan diperbaiki tanpa mengurangi kualitas kode. Salah satu cara

meningkatkan *maintainability* adalah refaktorisasi kode, yang merupakan proses mengubah kode tanpa mengubah perilaku eksternal. Refaktorisasi kode penting dalam meningkatkan kualitas perangkat lunak dan meningkatkan *maintainability* (Khaleel & Al-Khatouni 2023).

Dalam beberapa tahun terakhir, dalam pengembangan aplikasi terdapat beberapa potensi rendahnya tingkat pemeliharaan aplikasi. Pemeliharaan suatu aplikasi perlu dilakukan agar aplikasi tersebut tetap terjaga performanya. Penelitian sebelumnya, penerapan prinsip *clean code* dihubungkan dengan proses *refactoring*. *Clean code* adalah kode yang dirancang agar mudah dipahami dan dimodifikasi, serta mengikuti prinsip-prinsip desain yang baik. Kode yang mengikuti prinsip *clean code* lebih sedikit mengandung kesalahan dan lebih mudah dipelihara. Proses *refactoring* dilakukan untuk meningkatkan kualitas perangkat lunak dengan mengubah kode tanpa mengubah fungsionalitasnya. *Refactoring* membantu mengurangi kompleksitas kode, memperbaiki keterbacaan dan meningkatkan efisiensi pemeliharaan.

Dalam penelitian sebelumnya, dilakukan dengan menyediakan dataset *refactoring* yang terbuka dan berisi informasi *refactoring* yang diterapkan pada beberapa rilis sistem Java sumber terbuka, serta metrik kode sumber

terkait. Dataset ini mencakup informasi *refactoring* pada tingkat metode dan kelas, serta memungkinkan penelitian lebih lanjut tentang hubungan antara *refactoring* dan kualitas kode. Selain itu, penelitian ini menggunakan metode statistik untuk menyelidiki efek *refactoring* pada atribut kualitas kode internal (Arif & Rana 2020).

Studi kasus menggunakan sistem monitoring jurnal dipilih didasarkan pada kompleksitas code dan nilai *maintainability* website yang tergolong rendah, kompleksitas ini dapat meningkat seiring berjalannya waktu dan objek penelitian dalam studi ini adalah *website monitoring* sistem jurnal, yang merupakan aplikasi berbasis web yang dirancang untuk mengelola dan mengawasi aktivitas publikasi dan akreditasi jurnal akademis (Tampubolon et al, 2023). Sistem ini memainkan peran krusial dalam lingkungan akademis dengan menyajikan data yang terperinci tentang publikasi dan status akreditasi jurnal, yang memerlukan fitur-fitur kompleks dan interaksi yang terintegrasi (Agianto & Astuti, 2023). Website ini mengelola berbagai modul termasuk manajemen artikel, pembaruan data jurnal secara otomatis, serta fitur filter data. Seiring dengan berjalannya waktu, peningkatan jumlah data dan penambahan fitur sangat mungkin membuat struktur sistem menjadi

lebih rumit, sehingga seringkali menyebabkan kode menjadi kompleks dan sulit dipelihara seiring waktu. (Putra et al, 2022).

Komplesitas ini terutama terlihat pada interaksi antar modul yang semakin erat dan saling bergantung. Tanpa pemisalahan logika dan struktur kode yang baik, setiap perubahan kecil pada satu modul bisa berdampak pada modul lainnya. Hal ini tidak hanya memperlambat pengembangan fitur baru, tetapi juga meningkatkan risiko kesalahan dan menghambat proses pemeliharaan kode. Selain itu, penambahan data yang terus menerus memerlukan pemrosan yang akurat, hal ini menambah tantangan dalam memastikan sistem berjalan optimal, Dalam kondisi ini, tanpa penerapan prinsip *clean code*, sistem berpotensi mengalami masalah dalam hal keterbacaan dan pemeliharaan kode, membuat pengembang kesulitan dalam mengelola perubahan seiring bertambah kompleksnya sistem.

Dengan fokus pada optimasi *maintainability* aplikasi, penelitian ini menerapkan prinsip *clean code* untuk meningkatkan keterbacaan, mengurangi duplikasi, dan memperbaiki dokumentasi serta performa sistem (Lestari et al, 2024). Melalui penerapan *refactoring* berdasarkan prinsip *clean code*, penelitian ini bertujuan untuk

mengidentifikasi dan menyelesaikan masalah yang ada dalam kode sumber, sehingga meningkatkan kualitas dan pemeliharaan sistem secara keseluruhan (Manuaba et al, 2023). Dengan menerapkan prinsip *clean code* pada sistem ini, diharapkan dapat diidentifikasi dan diatasi masalah-masalah pemeliharaan seperti kode yang tidak konsisten, duplikasi kode, dan kurangnya dokumentasi. Masalah-masalah ini sering mempengaruhi kualitas dan *maintainability code*, membuat pemeliharaan dan pengembangan fitur baru menjadi tantangan (Rachmad et al, 2023).

Selain itu penelitian ini akan melakukan pengukuran indeks *maintainability* sebelum dan sesudah restrukturisasi kode kemudian membandingkan kedua nilai tersebut. Hasil penelitian menunjukkan bahwa restrukturisasi kode dapat secara signifikan meningkatkan indeks pemeliharaan sistem perangkat lunak. Hal ini menunjukkan bahwa restrukturisasi merupakan hal yang penting teknik penting untuk meningkatkan kualitas dan pemeliharaan kode dalam jangka panjang. (Luca Ardito et al., 2020)

Performa sistem dapat menurun jika tidak dilakukan *refactoring* yang tepat hal ini malah menyebabkan perubahan dari fungsionalitas sistem maupun memperlambat kerja

sistem. Analisis pada bagian kode seperti *Comments*, *Function*, dan *General* dalam *code smells* bertujuan untuk meningkatkan keterbacaan, mengurangi duplikasi, dan memperbaiki dokumentasi serta performa (Wiji Kurniawan et al, 2024). Dengan demikian untuk menunjukkan bagaimana penerapan prinsip *clean code* dapat meningkatkan *maintainability* dan kualitas kode, serta memberikan rekomendasi konkret untuk pengembangan perangkat lunak dan praktik pemeliharaan di masa depan.

Penelitian ini berfokus pada analisis kode sumber tanpa mengubah fitur-fitur yang ada. *BeautifulSoup 4* adalah kerangka kerja pengembangan aplikasi web yang menggunakan bahasa *Python* yang bertujuan mempercepat pengembangan web daripada membuatnya dari awal. Bahasa pemrograman *Python* adalah bahasa pemrograman *scripting open source* yang banyak digunakan dan dirancang khusus untuk pengembangan web, dengan kemampuan untuk disisipkan ke dalam *HTML* (Wiji Kurniawan et al., 2024). penelitian memberikan rekomendasi *prototype* kode sumber sesuai prinsip *clean code*, dengan fokus pada analisis *Comments*, *Function*, dan *General* dalam *code smells*.

Prinsip *clean code* sangat penting untuk menjaga sistem perangkat lunak yang berkualitas tinggi dan dapat dipelihara. Studi ini memperkuat pentingnya kepemilikan kode yang jelas dalam mencegah erosi perangkat lunak dan mengurangi biaya pemeliharaan. Dengan mengikuti prinsip *clean code* dan memastikan kepemilikan yang jelas, tim pengembang dapat meningkatkan *maintainability* dan kualitas keseluruhan kode mereka. Prinsip *clean code* menganjurkan penulisan kode yang sederhana, dapat dipahami, dan mudah dipelihara, yang pada akhirnya mengurangi biaya pemeliharaan dan penurunan performa perangkat lunak. (Arhandi et al., 2019).

Dalam konteks ini, Q.S Al-Baqorah (2):195 menjelaskan

وَأَنْفِقُوا فِي سَبِيلِ اللَّهِ وَلَا تُلْقُوا بِأَيْدِيكُمْ إِلَى التَّهْلُكَةِ وَأَحْسِنُوا إِنَّ اللَّهَ يُحِبُّ
الْمُحْسِنِينَ ﴿١٩٥﴾

"Dan belanjakanlah (hartamu) di jalan Allah dan janganlah kamu menjatuhkan dirimu sendiri ke dalam kebinasaan. Dan berbuat baiklah, sesungguhnya Allah menyukai orang-orang yang berbuat baik."

Ayat ini menekankan pentingnya berbuat baik dan menjaga diri dari kerusakan. Dalam konteks pengembangan perangkat lunak, ini dapat diartikan sebagai pentingnya menghabiskan sumber daya dengan bijak dalam mengembangkan dan memelihara kode,

sehingga tidak ada pemborosan atau kerusakan yang tidak perlu.

1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas, maka dapat diidentifikasi masalah sebagai berikut :

- a. Tingkat *Maintainability* perangkat lunak yang rendah mengalami kesulitan dalam pemeliharaan akibat code yang tidak terstruktur seperti prinsip *clean code*.
- b. Sistem monitoring jurnal mengalami masalah kompleksitas kode yang meningkat dan tingkat *maintainability* yang rendah, disebabkan oleh penerapan prinsip *clean code* yang kurang optimal.

1.3 Rumusan Masalah

Berdasarkan latar belakang penelitian diatas, maka dapat dirumuskan masalah yaitu :

- a. Bagaimana penerapan *Clean Code* dapat meningkatkan *Maintainability Website*?
- b. Bagaimana efisiensi & dan efektivitas *Clean Code* melalui *refactoring* dalam pemeliharaan kode program?

1.4 Batasan Masalah

Berdasarkan latar belakang penelitian yang telah dijelaskan, adapun batasan masalah dari penelitian ini adalah sebagai berikut:

- a. Penelitian ini hanya berfokus pada penerapan prinsip-prinsip clean code untuk meningkatkan Maintainability Indeks dari aplikasi Website Monitoring Sistem Jurnal Sebagai tugas akhir
- b. Pengukuran yang dilakukan dalam penelitian ini hanya mencakup analisis Maintainability Index yang dihitung dari metrik Line of code (LOC), Cyclomatic Complexity (CC), dan Helstead Volume (HV).
- c. Penelitian ini hanya membandingkan nilai Maintainability Indeks sebelum dan sesudah penerapan prinsip Clean Code menggunakan metode komperasi mean dan uji-t berpasangan (paired t-test) tanpa melakukan pengujian terhadap performa waktu respon aplikasi , konsumsi memori, atau performa server.

1.5 Tujuan Penelitian

Berdasarkan rumusan masalah diatas, berikut merupakan tujuan dari penelitian :

- a. Mengukur pengaruh penerapan prinsip *Clean Code* terhadap indeks *Maintainability* pada perangkat lunak memastikan bahwa kode tetap berkualitas tinggi serta mudah dipelihara.
- b. Meningkatkan efisiensi dan efektivitas pemeliharaan kode program melalui refaktorisasi, menganalisis bagaimana refaktorisasi dapat mengurangi kerumitan code, dan meningkatkan keterbacaan.

1.6 Manfaat Penelitian

a. Manfaat Teoritis

Adapun manfaat teoritis dari penelitian ini sebagai berikut :

1. Memperkaya literatur mengenai prinsip *Clean Code* dan *Maintainability* dengan menambahkan data empiris tentang dampak refaktorisasi terhadap kualitas kode program.
2. Analisis teknik refaktorisasi tertentu terhadap atribut code, memberikan kontribusi teoritis yang signifikan dalam bidang rekayasa perangkat lunak.
3. Mengevaluasi perubahan indeks *maintainability* sebelum dan sesudah refaktorisasi.

b. Manfaat Praktis

Adapun manfaat Praktis dari penelitian ini sebagai berikut :

1. Sebagai panduan bagi pengembang perangkat lunak tentang bagaimana menerapkan prinsip *clean code* dan refaktorisasi untuk meningkatkan kualitas dan *maintainability* kode program.
2. Dapat berkontribusi membantu pengembang mengurangi biaya dan waktu yang diperlukan untuk pemeliharaan perangkat lunak.
3. Membantu mengoptimalkan melalui refaktorisasi akan memiliki kinerja lebih baik, mengurangi *bug*, dan minim gangguan teknis.

1.7 Sistematika Penulisan

Sistematika penulisan ditujukan kepada pembaca agar lebih mudah dalam memahami isi laporan penelitian. Secara garis besar sistematika penulisan proposal ini terdiri dari :

BAB I PENDAHULUAN

Dalam bagian pendahuluan membahas latar belakang penelitian ini dimana untuk mengetahui sebab penelitian ini dilakukan dan selanjutnya membahas

rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian dan langkah penyelesaian.

BAB II LANDASAN PUSTAKA

Pada landasan pustaka membahas hal-hal dasar yang berisi teori-teori yang berkaitan dengan analisis prinsip menggunakan *Clean Code* dan hal-hal yang mendukung dalam hal melakukan analisis ini.

BAB III METODOLOGI PENELITIAN

Bagian ini membahas tentang sumber data dan tahapan yang dilakukan dalam pembuatan penelitian serta gambaran umum sistem yang akan dikerjakan.

BAB IV HASIL DAN PEMBAHASAN

Bagian ini menjelaskan tentang hasil pengolahan data, pengujian, serta menjelaskan kelebihan dan kekurangan dari hasil olahan data.

BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan yang merupakan rangkuman dari hasil penelitian ini dan berisi saran-saran yang membangun untuk pengembangan yang lebih baik di masa depan

BAB II

LANDASAN PUSTAKA

2.1 Kajian Pustaka

2.1.1 *Maintainability*

Menurut Rudolph Frederick Stapelberg, *maintainability* dari perspektif pemeliharaan adalah kemungkinan bahwa suatu item yang gagal akan dikembalikan ke kondisi operasional yang efektif dalam waktu tertentu. Sedangkan menurut Di Biase et al., (2019) *maintainability* didefinisikan sebagai kemudahan suatu sistem perangkat lunak untuk dimodifikasi dan dipelihara. Oleh karena itu, *maintainability* dapat diartikan sebagai kemampuan untuk melakukan perawatan terhadap suatu sistem dalam rentang waktu yang ditentukan ketika sistem tersebut mengalami masalah. Dalam ISO-25010, *maintainability* memiliki beberapa subfaktor yang menentukan apakah suatu sistem dapat dianggap *maintainable* jika memenuhi subfaktor tersebut. Berikut beberapa sub faktor *Maintainability* antara lain.

a. *Modularity*

Modularity menggambarkan sejauh mana suatu sistem atau program yang terdiri dari berbagai modul dirancang sedemikian rupa sehingga perubahan pada satu modul memiliki dampak minimal terhadap modul lainnya.

b. *Reusability*

Reusability menjelaskan sejauh mana suatu aset dapat digunakan kembali dalam lebih dari satu sistem atau dalam pembuatan aset lainnya.

c. *Analysability*

Analysability menggambarkan tingkat keefektifan dan efisiensi dalam menilai dampak perubahan yang dimaksudkan pada satu atau lebih modul dari suatu produk atau sistem. Ini juga mencakup kemampuan untuk mendiagnosis penyebab kegagalan atau mengidentifikasi bagian yang perlu dimodifikasi.

d. *Modifiability*

Modifiability menggambarkan sejauh mana suatu produk atau sistem dapat dimodifikasi

secara efektif dan efisien tanpa menimbulkan bug atau menurunkan kualitas produk yang ada.

e. *Testability*

Testability menggambarkan tingkat keefektifan dan efisiensi dalam menentukan apakah suatu sistem, produk, atau komponen memenuhi kriteria pengujian yang telah ditetapkan. (Mulyawan et al., 2021)

2.1.2 Maintainability Index

Menurut Terzimehic dalam makalah "Pembangunan Kakas Bantu Untuk Mengukur *Maintainability Index* Pada Perangkat Lunak Berdasarkan Nilai *Halstead Metrics* dan McCabe's *Cyclomatic Complexity*," *Maintainability Index* (MI) adalah metrik perangkat lunak yang mengukur seberapa mudah atau sulit suatu perangkat lunak dapat dipelihara atau diubah di masa depan. *Maintainability Index* dihitung berdasarkan nilai formula dari *Halstead's Volume* (HV), *Cyclomatic Complexity* (CC), dan *Lines of Code* (LOC). Semakin tinggi nilai dari kalkulasi *maintainability index* mengindikasikan bahwa kode program memiliki

tingkat *maintainability* yang baik, dimana hal ini akan membuat *source code* lebih mudah dipahami dan dirawat sehingga akan lebih mudah dalam pencarian dan perbaikan kecacatan (*bugs*) (Atmaja et al., 2019).

$$MI = 171 - 5.2 \times \ln(LOC) - 0.23 \times CC - 16.2 \times \ln(HV) \quad (2.1)$$

Keterangan :

MI: *Maintainability Index* (Indeks Kemampuan Pemeliharaan)

LOC: *Lines of Code* (Jumlah Baris Kode)

CC: *Cyclomatic Complexity* (Kompleksitas Siklomatik)

Halstead Volume: Volume Halstead

ln: Logaritma natural

Table 2. 1 Klasifikasi Maintainability Index

Klasifikasi	Nilai Maintainability
<i>High maintainability status, project</i>	MI>85

<i>included in good condition.</i>	
<i>Maintainability status, there are problems / problems with the project but not too serious.</i>	$65 < MI \leq 85$
<i>Low maintainability status, project has a critical or fatal technical error.</i>	$MI \leq 64$

Untuk dapat menghitung *maintainability index* terdapat beberapa tahapan yang harus dilakukan untuk mendapatkan nilai masukan yang nantinya akan digunakan untuk menghitung nilai *maintainability index*.

a. Halstead Metric

Halstead's Metric adalah pengukuran yang dikembangkan untuk mengukur kompleksitas

dalam suatu program langsung dari source code. Pengukuran dilakukan dengan menentukan ukuran kuantitatif kompleksitas dari operator dan operand dalam suatu modul sistem. Pada Halstead's metric terdapat enam jenis komponen yaitu :

1. *Length of Program*

Length of program adalah kalkulasi jumlah total operator dan operand yang muncul. Persamaan *Length of the program* dirumuskan pada persamaan.

$$N = N1 + N2 \quad (2.2)$$

Keterangan :

N1 = Jumlah Operator

N2 = Jumlah Operand

2. *Vocabulary of the program*

Vocabulary of the program adalah kalkulasi jumlah operator unik dan operand unik yang muncul dalam program. Berikut merupakan Persamaan *Vocabulary of the program* antara lain.

$$n = n1 + n2 \quad (2.3)$$

Keterangan :

n = *Vocabulary of program*

n_1 = Jumlah operator unik

n_2 = Jumlah operan unik

3. *Volume of the program*

Volume of the program adalah *volume* dalam *halstead metric* yang digunakan untuk mengetahui *volume* program. Berikut merupakan Persamaan *volume of the program* antara lain.

$$HV = N \times \log_2(n) \quad (2.4)$$

Keterangan :

V = *Volume of the program*

N = Nilai kalkulasi *length of the program*

n = nilai kalkulasi *Vocabulary of the program*

4. *Difficult*

Difficulty dalam *Hakstead Metric* digunakan untuk mengetahui kesulitan dan pengembangan program. Berikut Persamaan *difficulty* antara lain.

$$D = \frac{n_1}{2} \times \frac{N_2}{n_2} \quad (2.5)$$

Keterangan :

D = Difficulty

N2 = Total semua operand yang muncul

n1 = Jumlah operator unik yang muncul

n2 = jumlah *operand* unik

5. *Effort*

Effort dalam *halstead metric* di gunakan untuk mengetahui sumber daya yang digunakan untuk pengembangan program. Persamaan *effort* dirumuskan pada persamaan.

$$E = D \times V \quad (2.6)$$

Keterangan

E = *Effort*

D = nilai *difficulty*

V = *Volume of the program*

6. *Number of Bugs Expected in The Program*

Number of bugs expected in the program dalam *halstead metric* di gunakan untuk mengetahui prediksi bug pada program. Berikut Persamaan *Number of bugs expected in the program* antara lain.

$$B = \frac{V}{3000} \quad (2.7)$$

Keterangan :

B = Number of bugs expected in the program

V = Nilai kalkulasi Volume of the program

b. Cyclomatic Complexity

Cyclomatic complexity adalah sebuah metrik yang digunakan dalam rekayasa perangkat lunak untuk mengukur kompleksitas sebuah program berdasarkan jumlah jalur linier independen dalam kode program. Metrik ini diperkenalkan oleh Thomas J. McCabe pada tahun 1976, dengan tujuan untuk mengukur jumlah pengujian minimal yang dibutuhkan untuk memastikan bahwa semua jalur eksekusi di dalam suatu program telah diuji (Lenarduzzi et al., 2023). *Cyclomatic complexity* dihitung dengan menggunakan representasi graf dari program, di mana simpul mewakili pernyataan program dan tepi mewakili jalur kontrol. Menurut Khaleel & Al-Khatouni (2023) menghitung *cyclomatic complexity* terdapat rumus yaitu adalah:

$$CC = E - N + 2P \quad (2.8)$$

Keterangan :

CC = *Cyclomatic complexity*

E = Jumlah edge (sisi) dalam graf kontrol alur

N = Jumlah node (simpul),

P = Jumlah komponen terhubung dalam graf

Penggunaan *cyclomatic complexity* dapat membantu pengembang perangkat lunak untuk mengidentifikasi bagian-bagian kode yang lebih kompleks, yang mungkin lebih sulit diuji dan lebih rentan terhadap kesalahan. Semakin tinggi nilai *cyclomatic complexity*, semakin banyak jalur yang perlu diuji untuk menjamin bahwa semua skenario eksekusi telah tercakup. Sebagai contoh, kode dengan *cyclomatic complexity* yang rendah umumnya lebih mudah untuk dipelihara dan diuji dibandingkan dengan kode yang memiliki *cyclomatic complexity* tinggi (De Silva et al., 2023). Oleh karena itu, pengembang sering menggunakan metrik ini sebagai indikator dalam pengambilan keputusan mengenai *refactoring* kode atau pengoptimalan pengujian.

Cyclomatic complexity menjadi salah satu metrik yang penting dalam rekayasa perangkat lunak, terutama dalam konteks pengujian dan pemeliharaan kode. Meskipun memiliki

keterbatasan, metrik ini dapat memberikan wawasan berharga bagi pengembang untuk meningkatkan kualitas kode mereka melalui pengurangan kompleksitas yang tidak perlu dan peningkatan keterujian (Fenton & Bieman, 2014).

c. *Line of Code*

Line of Code (LOC) merupakan metrik dasar yang umum digunakan dalam rekayasa perangkat lunak untuk menilai ukuran dan kompleksitas perangkat lunak. Metrik ini menghitung total baris kode sumber dalam suatu program, termasuk kode yang dapat dieksekusi dan komentar. Meskipun sederhana, LOC berfungsi sebagai indikator awal untuk memperkirakan usaha, biaya, dan kualitas dalam pengembangan perangkat lunak (Khaleel & Al-Khatouni 2023).

Peningkatan jumlah baris kode sering kali menunjukkan bahwa proyek tersebut memerlukan waktu dan sumber daya yang lebih banyak untuk pengembangan. LOC juga sering diterapkan dalam konteks estimasi usaha dan

pengukuran produktivitas pengembang (Aswini & Yazhini, 2017).

Menurut Aswini & Yazhini (2017) LOC dapat dibagi menjadi beberapa jenis, yaitu:

1. *Physical Lines of Code* (LOC): Metrik ini menghitung semua baris kode sumber tanpa memperhitungkan isinya, seperti spasi kosong atau komentar. Kadang disebut juga sebagai *Non-Comment Lines of Code* (NCLOC) atau *effective Lines of Code* (eLOC).
2. *Logical Lines of Code* (LLOC): Metrik ini menghitung jumlah pernyataan dalam program. Sebagai contoh, di bahasa pemrograman C dan Java, pernyataan program diakhiri dengan tanda titik koma, meskipun aturan ini tidak berlaku untuk semua bahasa.
3. *Comment Lines of Code* (CLOC): Metrik ini menghitung jumlah baris komentar dalam kode, tanpa memperhitungkan kualitas dari komentar tersebut.
4. *Comment Rate* (CR): Rasio ini menghitung perbandingan antara CLOC dan LOC, memberikan indeks persentase dari

banyaknya komentar dalam program. Semakin tinggi rasio komentar (hingga batas tertentu), semakin mudah dipahami program tersebut. Rumus untuk menghitung CR adalah sebagai berikut:

$$CR = \left(\frac{CLOC}{LOC} \right) \times 100 \quad (2.9)$$

Keterangan :

CR = *Comment Rate*

CLOC = *Comment Lines of Code*

LOC = *Physical Lines of Code*

2.1.3 Prinsip *Clean Code*

Berdasarkan buku *Clean Code: A Handbook of Agile Software Craftsmanship* karya Robert C. Martin, *clean code* adalah kode yang mudah dibaca, mudah dipahami, dan mudah dipelihara. Kode ini ditulis dengan tujuan agar orang lain yang membacanya dapat mengerti dengan jelas apa yang dilakukan oleh program tersebut tanpa perlu penjelasan tambahan. *Clean code* berfokus pada kesederhanaan, dengan menghindari kompleksitas yang tidak perlu. Kode harus bebas dari duplikasi, memiliki struktur yang jelas, nama variabel dan fungsi yang deskriptif, serta mengikuti prinsip

SOLID agar mudah diuji dan dikembangkan di masa depan. Sementara menurut Grady, *clean code* itu simpel dan dibuat dengan baik seperti prosa. *Clean code* tidak pernah mengubah desain, tetapi membuat kode lebih mudah dikendalikan. Jadi, dapat disimpulkan bahwa *Clean Code* adalah metode untuk memperbaiki suatu program yang strukturnya sulit dipahami oleh pengembang lain. Menurut Robert C. Martin dalam bukunya *Clean Code: A Handbook of Agile Software Craftsmanship*, Ada beberapa faktor yang menentukan apakah suatu kode dapat disebut sebagai *clean code*.

a. *Meaningful Names*

Meaningful Names adalah prinsip-prinsip untuk memberi nama kepada kelas, variabel, dan fungsi agar lebih mudah dipahami oleh pengembang lain. Berikut adalah beberapa pedoman dalam menerapkan *meaningful names*.

1. Use Intention-Revealing Names

Use Intention-Revealing Names

memperhatikan pentingnya memberi nama variabel, fungsi, atau kelas dengan cara yang menjelaskan tujuannya secara langsung. Jika nama tersebut masih memerlukan komentar tambahan untuk menjelaskan maknanya, maka nama tersebut mungkin tidak cukup menggambarkan tujuan dari variabel, fungsi, atau kelas tersebut.

2. Avoid Disinformation

Avoid Disinformation untuk menghindari penggunaan nama yang menyesatkan, seperti menggunakan nama platform atau variasi *Unix*. Juga, disarankan untuk tidak menggunakan variasi kecil yang sulit dibedakan, seperti huruf "l" kecil dan "O" besar, terutama dalam kombinasi sebagai nama variabel.

3. Use Pronounceable Names

Use Pronounceable Names menyatakan pentingnya menggunakan nama yang dapat diucapkan oleh manusia sehingga mudah dipahami tujuannya.

4. Use Searchable Name

Use Searchable Name adalah penggunaan nama yang mudah dicari. Penggunaan nama-nama seperti huruf tunggal atau konstanta numerik dapat menyulitkan pencarian mereka di seluruh kode. Contoh buruk termasuk penggunaan huruf "e" sebagai nama variabel, karena kata "e" adalah salah satu kata yang paling umum dalam bahasa Inggris. Pastikan nama yang digunakan tidak hanya bermakna, tetapi juga dapat dicari dengan mudah.

5. Avoid Mental Mapping

Avoid Mental Mapping menghindari masalah dengan penamaan variabel menggunakan satu huruf. Dalam konteks looping, seperti penggunaan huruf i, j, atau k, masalahnya tidak terlalu besar jika lingkungannya kecil. Namun, dalam konteks lain, menggunakan nama variabel yang hanya satu huruf adalah pilihan yang buruk.

6. Class Names

Class names menyarankan bahwa nama kelas dan objek harus menggunakan frasa kata benda seperti *Customer*, *WikiPage*, dan

Account. Hindari menggunakan kata seperti *Manager, Processor, Data*, atau *Info* untuk nama kelas. Nama kelas tidak boleh mengandung kata kerja, dan huruf pertama nama kelas harus kapital.

7. *Don't Pun*

Don't Pun menyarankan agar pengembang menghindari penggunaan kata yang memiliki makna yang sama tetapi sintaksnya berbeda dalam konteks yang sama. Sebagai contoh, jika kata "*add*" sudah sering digunakan dalam sebuah kelas, sebaiknya digunakan sintaks lain seperti "*insert*" atau "*append*", meskipun penulisan keduanya berbeda secara sintaksis, tetapi memiliki makna yang sama dengan "*add*".

8. *Don't be Cute*

Don't be Cute mengatakan bahwa kita sebaiknya tidak menggunakan kata-kata yang terkesan lucu atau menggemaskan. Sebaliknya, kita harus menggunakan kata-kata yang sesuai dan jelas menyampaikan apa yang diharapkan.

9. *Method Names*

Method Names maksudnya merupakan nama metode harus memiliki nama dari kata kerja atau frasa kata kerja seperti *postpayment*, *deletePage*.

b. Functions

1.Small Function

Small function menyarankan agar fungsi dibuat sesingkat dan sesederhana mungkin . Fungsi yang kecil lebih mudah dipahami dan diuji. Jika suatu fungsi memiliki lebih dari satu tanggung jawab , itu tanda bahwa fungsi tersebut perlu dipecah menjadi beberapa fungsi yang lebih kecil. Setiap fungsi harus hanya melakukan satu tugas tertentu .

2.Do one Thing

Fungsi yang bersih harus hanya melakukan satu hal saja . Jika suatu fungsi tampaknya melakukan beberapa tugas atau memiliki terlalu banyak logika maka fungsinya perlu dipisahkan .

3.Avoid side Effect

Fungsi yang baik tidak boleh memiliki efek samping yang tidak terduga . Fungsi harus mengembalikan hasil yang diharapkan

berdasarkan input yang diberikan tanpa mempengaruhi kondisi eksternal lain dalam program.

4. Use Command Query Separation

Prinsip ini menyatakan bahwa sebuah fungsi sebaiknya melakukan salah satu dari dua hal : mengubah status objek (perintah/*command*) atau mengembalikan data tanpa mengubah apapun (*query*). Kombinasi keduanya dalam satu fungsi sering kali menimbulkan kebingungan. Sebagai contoh sebuah fungsi bernama `reduce_stock (product, quantity)` yang hanya mengurangi stok tanpa mengembalikan nilai `get_stock(product)` yang hanya mengembalikan jumlah stock tanpa mengubah apapun . Jadi aksi dan query dipisah , sehingga masing masing fungsi memiliki satu tanggung jawab yang jelas .

5. No Argument of few Argument

Function yang ideal tidak menggunakan argumen sama sekali . Namun, jika memang diperlukan , maka sebaiknya jumlah argumen dibatasi seminimal mungkin, dengan

rekomendasi tiga argumen. Penggunaan terlalu banyak argument dapat membuat fungsi sulit dipahami dan digunakan .

6. Avoid Flag Argumen

Function yang memiliki parameter boolean (*flag argument*) sering kali menunjukkan bahwa fungsi tersebut melakukan lebih dari satu tugas . Sebaliknya , fungsi dipecah menjadi dua atau lebih fungsi terpisahkan berdasarkan nilai boolean tersebut.

7. Dry principle (don't repeat yourself)

Jangan menduplikasi kode diberbagai tempat. Jika ada fungsi yang menggunakan logika yang sama dengan fungsi lain, sebaiknya logika tersebut diubah menjadi satu fungsi terpisah yang bisa dipanggil oleh fungsi lainnya.

c. Code Duplication

Don't Repeat Yourself (DRY) atau pengulangan kode adalah salah satu tanda kode yang buruk . Kode yang bersih meminimalkan pengulangan dengan memanfaatkan abstraksi yang sesuai . Misalnya, logika digunakan di beberapa

tempat sebaiknya diubah menjadi fungsi atau class yang dapat dipanggil atau digunakan kembali

d. Comment

Comment dalam kode sebaiknya digunakan hanya jika diperlukan untuk menjelaskan alasan *why* , bukan *what*, karena kode yang baik seharusnya sudah cukup jelas dengan penamaan variabel dan struktur yang baik . Komentar yang berlebih atau menjelaskan hal yang sudah jelas melalui kode hanya akan membuat kode sulit dipahami dan tidak perlu. Idealnya, kode seharusnya *Self-Explanatory* tanpa memerlukan komentar tambahan, namun jika ada bagian yang kompleks, komentar bila digunakan untuk menjelaskan konteks atau keputusan desain tertentu yang tidak terlihat dari kodenya sendiri.

e. Formating

1. Consistent Formating

Kode yang baik harus mengikuti pola penulisan atau format yang konsisten

diseluruh bagian program. Struktur seperti identasi, spasi dan pemisahan baris harus diterapkan secara seragam sehingga kode mudah dibaca.

2. Vertical and horizontal Formatting

Keterbacaan kode juga dapat ditingkatkan dengan menjaga panjang baris dan struktur blok kode yang rapi secara vertikal. Blok kode yang terkait harus dikelompokkan, dan fungsi-fungsi yang saling berhubungan harus diletakan berdekatan.

f. Error handling

1. Use Exceptions, Not Return codes

Penanganan error harus dilakukan dengan menggunakan exception alih-alih dengan return, memisalhkan logika utama dari penanganan error, membuat kode lebih rapi dan fokus pada tugas utamanya.

2. Dont pass null

Hindari mengirimkan null sebagai argument ke function. Jika seatu argumen bisa null, fungsi tersebut sering kali perlu menambahkan banyak pengecekan untuk menghindari masalah. Hal ini menyebabkan

kode menjadi rumit susah di pelihara dan meningkatkan risiko *NullPointerException*. Sebaiknya, pastikan nilai yang dikitimi selalu valid atau gunakan objek default sebagai pengganti null.

3. Don't return Null

Kode yang baik menghindari mengembalikan null dari function. Ketika null dikembalikan, kode yang memanggil function harus menambahkan pengecekan null, yang dapat menambah kompleksitas dan membuat kode lebih rentan terhadap error. Sebaiknya, kembalikan objek kosong atau nilai default.

g. Object and data struktur

Dalam pemrograman berorientasi objek, penting untuk menjaga pemisahan antara data dan logika yang memanipulasinya. Setiap object harus menyembunyikan detail implementasi dan hanya menampilkan antarmuka yang sederhana untuk digunakan, Data objek sebaiknya diakses melalui metode atau fungsi, bukan secara langsung, guna

menjaga enkapsulasi dan mengurangi kompleksitas.

1. Law of Demeter (LoD)

Biasa dikenal sebagai prinsip least knowledge, merupakan panduan dalam pemrograman berorientasi objek yang menyarankan agar objek hanya berbicara dengan “tetangganya” atau objek yang terkait langsung dengannya. Prinsip ini bertujuan untuk mengurangi keterkaitan (coupling) antar objek, sehingga kode lebih modular, mudah dipelihara dan lebih tahan terhadap perubahan. Sederhanya *law of demeter* sebagai berikut :

- a. berkomunikasi dengan dirinya sendiri (metode atau atribut internal)
- b. berkomunikasi dengan objek yang dimiliki secara langsung.
- c. berkomunikasi dengan objek yang dibuat langsung
- d. berkomunikasi dengan objek global yang tersedia dalam konteks .

h. classes

1. Single Responsibility Principle (SRP).

Setiap kelas harus memiliki satu tanggung jawab dan satu alasan untuk berubah. Kelas yang melakukan lebih dari satu hal lebih sulit dipelihara dan diuji

2. Cohesion and coupling

Kelas yang baik memiliki *cohesion* yang tinggi, artinya semua metode dalam kelas tersebut saling berhubungan dan relevan dengan tanggung jawab kelas . Sebaliknya, *coupling* antar kelas harus serendah mungkin, artinya kelas tidak boleh tergantung pada kelas lainnya .

2.1.4 Smell code

Code smells adalah istilah yang pertama kali diciptakan oleh Kent Beck dan Martin Fowler untuk menggambarkan indikasi bahwa ada sesuatu yang tidak beres dalam kode, yang mungkin menunjukkan masalah desain yang lebih mendalam. *Code smells* tidak selalu berarti bahwa ada bug dalam kode, tetapi bisa menjadi tanda bahwa kode tersebut memerlukan *refactoring* untuk meningkatkan kualitas dan *maintainability*.

- a. *Duplicate Code*. Merupakan bagian code yang sama atau hampir sama muncul di beberapa

tempat dalam aplikasi, masalah yang timbul akan memperbesar kemungkinan *bug*, misalnya apabila terdapat dua fungsi yang berbeda memiliki *logic* yang identik. Dengan refactor kode dapat menggunakan fungsi atau metode yang dipanggil ulang (M. Fowler, 2018).

- b. *Long Method*. Metode ini untuk sebuah fungsi yang menangani terlalu banyak tugas. Pemecahan metode menjadi beberapa metode kecil yang masing-masing melakukan suatu tugas spesifik. Metode ini juga untuk kode yang memiliki terlalu banyak baris (M. Feathers, 2020).
- c. *Large Class*. Kelas yang memiliki terlalu banyak metode atau atribut menjadi sulit dipahami, dipelihara dan diuji. Dengan memisahkan kelas besar menjadi beberapa kelas kecil dengan tanggung jawab masing-masing kelas yang lebih spesifik. Misalnya kelas yang menangani banyak fungsi yang tidak terkait (F. N. Topuz, 2022).
- d. *Long Parameter List*. Metode yang memiliki terlalu banyak parameter sehingga menjadi kesulitan untuk dipahami dan digunakan.

Fungsi terkadang membutuhkan lebih dari 3-4 parameter (S. Boutaib., dkk, 2021).

- e. *Divergent Change*. Kelas yang sering diubah karena alasan yang berbeda menunjukkan tanggung jawab kelas yang terlalu luas. Misalnya kelas yang menangani logika bisnis dan antarmuka pengguna. Solusinya dengan memisahkan kelas menjadi beberapa kelas dengan tanggung jawab tunggal (S. Boutaib., dkk, 2021).
- f. *Shotgun Surgery*. Suatu perubahan memerlukan banyak modifikasi kecil secara mendetail di berbagai kelas sehingga bagian masalah yang sulit untuk mengimplementasikan perubahan dapat diselesaikan dan mengurangi peningkatan resiko *bug*. Misalnya modifikasi fitur yang mengharuskan perubahan pada banyak modul yang berbeda, namun masalah ini dapat diselesaikan dengan konsolidasikan tanggung jawab terkait satu kelas atau modul (F. N. Topuz, 2022).
- g. *Feature Envy*. Metode dalam satu kelas yang terlalu tertarik pada data kelas lain. Metode ini

menyelesaikan ketergantungan yang tidak sehat antar kelas misalnya metode yang sering memanggil metode lain dari kelas yang berbeda. Masalah tersebut dapat diselesaikan dengan memindahkan metode atau logika kedalam kelas yang memiliki data yang diakses (F. N. Topuz, 2022).

- h. *Data Clumps*. Sekelompok data yang sering muncul bersama misalnya beberapa parameter yang selalu muncul bersama di beberapa tempat. Masalah yang muncul menunjukkan adanya kelas atau objek yang tersembunyi, solusinya dengan membuat objek baru untuk mengelompokkan data terkait (S. Boutaib., dkk, 2021).
- i. *Primitive Obsession*. Metode ini menggunakan tipe data primitif untuk segala sesuatu, membuat tipe yang lebih sesuai. Metode ini menurunkan *readability* dan meningkatkan risiko kesalahan. Contoh *primitive obsession* adalah apabila menggunakan integer untuk mempresentasikan berbagai berbagai jenis nilai. Solusinya dengan membuat objek baru untuk mengelompokkan data terkait.

- j. *Lazy Class*. Kelas yang hampir tidak memiliki fungsi atau data yang menambah kompleksitas tanpa manfaat yang jelas. Misalnya kelas yang hanya memiliki satu atau dua metode sederhana. Masalah tersebut dapat diselesaikan dengan menggabungkan kelas ke dalam kelas yang lebih besar dan relevan (S. Boutaib., dkk, 2021).

2.1.5 *Library BeautifulSoup 4*

BeautifulSoup adalah pustaka *Python* yang berfokus pada *parsing* HTML, dengan keunggulan dalam penggunaan memori sedang dan penggunaan CPU yang rendah. Hal ini membuat BeautifulSoup menjadi pilihan efisien untuk tugas scraping yang mengutamakan kecepatan dan efisiensi sumber daya. Pustaka ini unggul dalam ekstraksi data dari dokumen HTML dan menawarkan API yang sederhana serta intuitif, menjadikannya sangat cocok untuk proyek-proyek kecil atau kasus di mana solusi yang ringan diinginkan. (Y. Dimilintas., dkk, 2023).

2.2 Landasan Teori

2.2.1 *Maintainability*

- a. Implementasi *Clean Architecture* Dalam Membangun Aplikasi Mobile Menggunakan *Flutter*

Clean Architecture adalah konsep dalam pengembangan perangkat lunak yang bertujuan untuk menciptakan struktur perangkat lunak yang sederhana, mudah diuji, dan tidak bergantung pada detail implementasi.

Dalam penelitian ini penerapan prinsip *Clean Architecture* yang mempunyai tingkat *Maintainability* yang lebih tinggi daripada penerapan arsitektur MVC. Hasil perbandingan menunjukkan *maintainability* mencapai 78.2% dan arsitektur MVC hanya mencapai 54.8%, *maintainability* mendapatkan nilai lebih dari 67%.

Pengembangan aplikasi dengan *Clean Architecture* menghasilkan kode berkualitas tinggi yang mudah dikelola untuk pengembangan di masa depan. Selain itu, pendekatan ini memberikan efisiensi waktu

dan mengurangi beban kerja selama proses pengembangan, tanpa mengorbankan kualitas kode yang dihasilkan (W. P. Laksono., dkk 2024).

b. Analisis Penggunaan Pemograman Berorientasi Obyek Terhadap *Maintainability* Perangkat Lunak Menggunakan ODOO

ODOO adalah framework yang terkenal untuk pengembangan ERP karena kemudahannya dalam konfigurasi dan adaptasi. Penggunaan Object-Oriented Programming (OOP) dalam ODOO berpotensi meningkatkan kualitas dan kemudahan pemeliharaan kode.

Pada penelitian ini mengukur *maintainability* dalam proyek ERP yang dibangun dengan ODOO penting untuk mengidentifikasi area yang memerlukan perbaikan dalam desain dan implementasi. Penggunaan metrik objek-orientasi (OO metrics) dan submetric spesifik memberikan wawasan berharga tentang *maintainability*, termasuk kemudahan analisis, pengujian, dan perubahan kode. Penelitian ini bertujuan untuk

mengukur efektivitas penggunaan OOP dalam mempengaruhi kemudahan pemeliharaan sistem ERP ODOO, dengan menggunakan submetric dalam OO metrics untuk memahami bagaimana aspek-aspek tertentu dari desain dan implementasi OOP mempengaruhi *maintainability*.

Secara keseluruhan, penerapan prinsip desain yang baik seperti *Single Responsibility Principle* dapat secara signifikan meningkatkan kualitas dan kemudahan pemeliharaan kode dalam jangka panjang, menjadikan proses pengembangan dan pemeliharaan lebih efisien dan dapat diandalkan (M. H. Azhizi & M. A. Yaqin, 2024).

c. *A Systematic Literature Review of Machine Learning Techniques for Software Maintainability Prediction*

Maintainability dalam rekayasa perangkat lunak adalah atribut kualitas fundamental yang menantang untuk diprediksi, dengan penelitian menunjukkan bahwa model ensambel lebih akurat

dibandingkan model individu dalam memprediksi kemudahan pemeliharaan perangkat lunak berbasis objek.

Maintainability adalah salah satu atribut kualitas eksternal fundamental dalam rekayasa perangkat lunak dan sering diukur secara tidak langsung melalui atribut internal seperti kohesi kelas. Prediksi *maintainability* perangkat lunak menggunakan model prediksi yang akurat sangat penting untuk membantu organisasi dalam mengelola biaya, mengalokasikan sumber daya, dan merencanakan pemeliharaan yang efektif, terutama dalam sistem berbasis objek yang banyak digunakan.

Literature review ini mengidentifikasi dan menganalisis pengukuran, metrik, dataset, metode evaluasi, masalah machine learning, model prediksi individu, dan model ensambel dalam prediksi *maintainability* perangkat lunak melalui 56 studi yang dipublikasikan antara tahun 1991 dan 2018 (H. Alsolai & M. Roper, 2020).

2.2.2 Prinsip Clean Code

a. Implementasi *Clean Code* Pada Pengembangan Aplikasi Berbasis Web

Clean code adalah kumpulan kode yang mudah dibaca, dimengerti, dan di-maintenance, sehingga memudahkan komunikasi dan pemahaman antara pengembang. Menerapkan *konsep Clean Code* sangat penting dalam pengembangan aplikasi, terutama untuk kolaborasi tim. Setiap programmer harus mengikuti konsep clean code agar tidak mengalami kesulitan saat melanjutkan kode dari programmer lain. Selain itu, penerapan clean code dapat menghemat waktu dan biaya saat melakukan maintenance sistem.

Dalam penelitian ini memahami cara menerapkan clean code dalam pengembangan aplikasi web berbasis PHP. Diharapkan penelitian ini dapat bermanfaat bagi mereka yang ingin menerapkan konsep clean code dalam proyek pengembangan aplikasi (F. I. Inayah & M. Idris, 2021).

b. Pemograman Programmable *Logic Controller* (PLC) Berbasis “*Clean Code*”

PLC merupakan program jarak jauh yang tergantung dan terkoneksi melalui jaringan internet (IoT). Pemrograman PLC dengan pendekatan *clean code* menghasilkan program ladder PLC yang terstruktur dan sederhana dalam penyusunannya. Hal ini membuatnya tampak elegan dan nyaman dipandang, sehingga mudah dipahami oleh programmer PLC lainnya.

Pemrograman PLC berbasis *Clean Code* mempunyai 3 unsur penting dalam aspek pemrograman dan teknik, yaitu: Terstruktur, Rapi dan Teratur sehingga pemrograman PLC berbasis *Clean Code* sangat layak untuk dijadikan cara pemrograman ladder PLC dari tingkat dasar hingga tingkat lanjut (Y. B. Pamungkas, 2019).

2.2.3 SonarQube

SonarQube merupakan salah satu alat analisis kode statis sumber terbuka yang digunakan baik di dunia akademis maupun industri. SonarQube dapat dijalankan sebagai layanan di platform

SonarCloud atau diinstall pada server pribadi. Alat ini memverifikasi kepatuhan kode terhadap serangkaian aturan pemrograman yang telah ditetapkan untuk berbagai bahasa pemrograman. Jika Kode yang dianalisis melanggar aturan tertentu, SonarQube akan menandainya sebagai issue dan mengklasifikasikannya dalam kategori Reliability(bugs), Maintainability (Code Smell), dan Security (Vulnerabilities)

Menurut penelitian oleh Lenarduzzi et al.(2023), SonarQube digunakan oleh lebih dari 85.000 organisasi dan hampir 15.000 proyek sumber terbuka. Alat ini membantu pengembang dalam menjaga kualitas kode dengan mengidentifikasi dan melaporkan berbagai masalah yang dapat mempengaruhi keterbacaan dan pemeliharaan kode. Dengan mendeteksi code smell dan potensi kerentanan keamanan, SonarQube memastikan bahwa kode yang ditulis tetap bersih dan sesuai dengan standar pengembangan perangkat lunak yang baik (*Clean Code*).

2.2.4 Module Randon

Radon adalah *Library* python yang digunakan untuk menganalisis kompleksitas kode secara statis. Library ini menyediakan berbagai metrik seperti Cyclomatic Complexity (CC) dan Halstead Metrics yang membantu pengembang mengevaluasi kualitas kode mereka. Radon dapat dijalankan langsung pada file python atau diterapkan dalam proyek skala besar untuk mengukur tingkat kompleksitas kode yang mempengaruhi keterbacaan dan Maintainability perangkat lunak.

Menurut dokumentasi resmi Radon, CC dalam radon digunakan untuk mengidentifikasi bagian kode dengan percabangan yang terlalu kompleks, yang dapat meningkatkan risiko kesalahan dan menyulitkan *maintance*. Selain itu, Halstead Metrics mengevaluasi kompleksitas logis suatu program berdasarkan jumlah operator dan operand yang digunakan.

Radon sering digunakan dalam pengembangan perangkat lunak untuk memastikan bahwa kode tetap sederhana dan mudah dipahami sesuai dengan prinsip *Clean Code*. Dengan analisis berbasis metrik ini, pengembang dapat

mengidentifikasi bagian kode yang perlu dioptimalkan sehingga lebih efisien dan lebih mudah dipelihara.

2.3 Perbandingan Penelitian

Tabel 2.2 Perbandingan Penelitian

No	Artikel Penelitian	Metode Penelitian	Hasil Penelitian	Perbedaan
1.	Implementasi <i>Clean Code</i> pada Pengembangan Aplikasi Berbasis Web (Inayah & Idris, 2021)	Penelitian dimulai dengan mempelajari konsep <i>clean code</i> dan <i>guidelines</i> perusahaan. Implementasi melibatkan penamaan variabel yang sesuai, menghindari duplikasi	adanya konsep <i>clean code</i> , programmer dapat saling memahami maksud dan tujuan dari kode yang dituliskan, karena kode yang dituliskan mudah dibaca dan	"Optimasi <i>Maintainability</i> " mengkaji penerapan <i>Clean Code</i> untuk memperbaiki aplikasi yang ada, sedangkan

	kode, dan dipahami.	"Implementasi
	memastikan	ntasi
	kualitas kode	Clean
	menggunakan	Code"
	n tools	membah
	seperti	as
	SonarLint	penerapa
	dan PHP CS	n prinsip
	FIXER.	Clean
		Code
		dalam
		pengemb
		angan
		aplikasi
		web
		secara
		umum.

2.	Pengukuran Kualitas Website Pengurus Cabang NU Depok Menggunakan	Pengukuran kualitas aplikasi pada website Pengurus Nahdlatul Ulama Depok	Berdasarkan hasil pengukuran kualitas website pada <i>maintainability index</i> terdapat	Perbedaan dari penelitian yang dilakukan adalah pada
----	--	--	--	--

kan	melibatkan	potensi	objek
<i>Software</i>	analisis	rendahnya	penelitian
<i>Metric</i>	modul	tingkat	n serta
(Wijaya &	website,	pemeliharaan	variabel
Karmilasar	instalasi	dalam proses	penelitian
i, 2021)	sistem	pengembangan	n
	informasi	n website.	
	dan		
	PHPMetrics,		
	serta analisis		
	hasil laporan		
	dari		
	PHPMetrics.		

3.	Penerapan	Penelitian	Hasilnya	Perebeda
	<i>Clean Code</i>	dimulai	bukan hanya	an dari
	Pada	dengan studi	mengurangi	penelitian
	Aplikasi	literatur,	kompleksitas	n yang
	Siizin	diikuti	dan	dilakuka
	Berdasark	wawancara	meningkatkan	n adalah
	an <i>Smells</i>	tentang	keterbacaan	terletak
	<i>And</i>	<i>source code</i> ,	kode tetapi	pada
	<i>Heuristics</i>	identifikasi	juga	fokus dan
		fitur, analisis	memberikan	tujuan
		<i>Code Smells</i> ,	manfaat dalam	penelitian

<i>(Widhiarso et al, 2024)</i>	dan diakhiri dengan implementasi <i>refactoring</i> untuk memperbaiki kode aplikasi sesuai prinsip <i>Clean Code</i> .	pemeliharaan, pengembangan dan kolaborasi tim. Setelah refactor kode tersebut, aplikasi SIIZIN telah memenuhi standar <i>clean code</i> terutama pada kategori <i>names, comments dan generals</i> .	n penelitian ini yaitu penerapannya <i>clead code</i> pada aplikasi berdasar kan <i>Smells and Heuristic</i> .
--------------------------------	--	--	---

BAB III

METODE PENELITIAN

Pada penelitian ini menggunakan metode kuantitatif komparatif mean untuk menganalisis perbedaan tingkat *maintanability* sebelum dan sesudah penerapan prinsip clean code. Analisis dilakukan dengan menghitung nilai rata-rata dari metrik Line of Code (LOC), Cyclomatic Compaxity (CC), Halstead Volume (HV), dan Maintainability Indeks (MI) guna memperoleh gambaran mengenai efektivitas refactorisasi kode dalam meningkatkan maintainability.

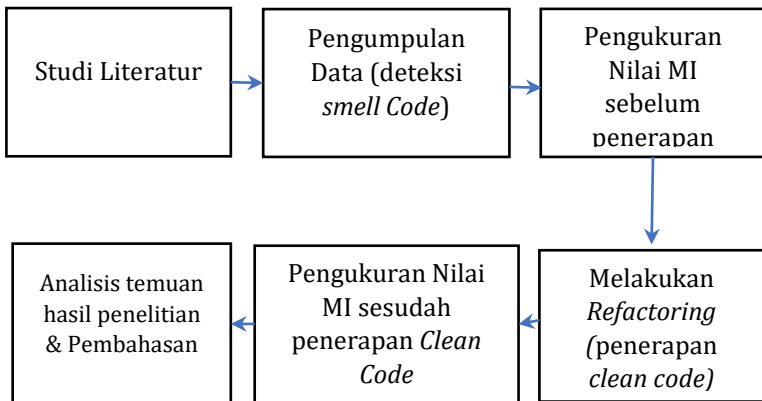
Selain itu, penelitian ini juga menggunakan uji-t berpasangan (paired t-test) untuk mengevaluasi signifikansi perbedaan antara maintainability indeks sebelum dan sesudah refactoring. Pengujian ini dilakukan untuk menentukan apakah terdapat perubahan signifikan dalam maintainability setelah penerapan prinsip clean code . Pengujian ini dilakukan dengan *microsoft excel*, menggunakan fitur *Data Analysis ToolPak*. Kriteria pengujian ditetapkan dengan $\alpha = 0.05$. Jika T Stat (t hitung) < t critical two-tail (t-tabel) maka, H_0 ditolak dan sebaliknya.

Adapun hipotesis dalam penelitian ini sebagai berikut:

1. Hipotesis Penelitian

- a. H_0 : tidak terdapat perbedaan signifikan pada nilai maintainability indeks sebelum dan sesudah penerapan *clean code*
- b. H_1 : terdapat perbedaan signifikan pada nilai maintainability indeks sebelum dan sesudah penerapan *clean code*.

3.1 Alur Kerja Penelitian



Gambar 3. 1 Alur kerja Penelitian

Pada penelitian ini menggunakan metode kuantitatif.

Adapun tahapan penelitian sebagai berikut :

3.1.1 Studi literatur

penelitian ini dimulai dengan studi literatur yang mendalam untuk memahami prinsip-prinsip dasar *clean code* dan pentingnya *maintainability* dalam pengembangan perangkat lunak. Langkah awal ini penting untuk membangun dasar pengetahuan yang solid sebelum melanjutkan ke tahapan berikutnya. Serta memperdalam pemahaman hubungan antara *prinsip clean code* dengan *smell code*.

3.1.2 Pengumpulan Data

Setelah memiliki pemahaman yang mendalam mengenai prinsip *clean code*, dilakukannya pengumpulan data dengan identifikasi *smell code* menggunakan tools SonarQube untuk mendeteksi *code smell* yang ada didalam kode. Data *Smell code* yang ditemukan nantinya akan di *refactoring* secara manual sesuai dengan prinsip *clean code*.

3.1.3 Pengukuran Nilai MI sebelum penerapan *Clean code*

Pada tahapan ini menghitung nilai beberapa metrix diantaranya : *Line of code* (LOC), *Cyclomatic Complexity* (CC), *Halstead Volume* (HV) sehingga

dapat dihitung nilai *Maintainability Index* sebelum diterapkannya prinsip *clean code*. Data ini akan menjadi data pembanding dalam evaluasi hasil *refactoring* yang akan digunakan .

3.1.4 Melakukan Refactoring.

Berdasarkan hasil pengukuran nilai *maintainability index* dan pengumpulan data *smell code* menggunakan sonarqube selanjutnya dilakukannya *refactoring code* secara manual berdasarkan prinsip *clean code*. Mulai dari perubahan nama variabel dan fuction yang lebih deskriptif , perubahan code yang terlalu kompleks, mengurangi duplikasi kode, penulisan format sesuai dengan prinsip *clean code*, dan konsistensi format penulisan ,sehingga kode yang dituliskan sesuai dengan prinsip *clean code*. Selama proses *refactoring*, dilakukan pengujian fungsionalitas untuk memastikan bahwa setiap modifikasi tidak mengganggu fungsionalitas aplikasi .

3.1.5 Pengukuran Nilai MI sesudah penerapan Refactoring

Setelah melakukan *refactoring code* berdasarkan prinsip *clean code*, langkah selanjutnya melakukan pengukuran nilai metrix

Line of code (LOC), *Cyclomatic Complexity* (CC), dan *Halstead Volume*(HV) sehingga dapat dihitung nilai *maintainability* indeks setelah diterapkannya prinsip *clean code*.

3.1.6 Analisis temuan dan hasil .

Berdasarkan hasil dari tahapan sebelumnya didapatkan nilai *Maintainability Index*, *Line of code* (LOC), *Cyclomatic Complexity* (CC), dan *Halstead Volume* (HV) sebelum dan sesudah *refactoring code* menggunakan prinsip *clean code*. Data hasil ini akan dianalisis menggunakan metode komperatif mean. Metode komperatif mean digunakan untuk membandingkan hasil pengukuran sebelum dan sesudah *refactoring*. Nilai rata-rata dari *Maintainability Index*, *Line of code* (LOC), *Cyclomatic Complexity* (CC), dan *Halstead Volume* (HV) dibandingkan untuk mengidentifikasi peningkatan *maintainability*. Analisis ini akan menunjukkan sejauh mana penerapan prinsip *clean code* berhasil meningkatkan *maintainability*, mengurangi kompleksitas kode serta meningkatkan keterbacaan kode.

3.2 Subjek Penelitian

Subjek Penelitian ini dilakukan berdasarkan Tugas Akhir Website monitoring jurnal. *Maintainability* yang diterapkan pada Website monitoring sistem jurnal dengan prinsip *Clean Code* dengan data *sampling*. Data *sampling* dalam optimasi *maintainability* dengan prinsip *clean code* diperlukan untuk memastikan evaluasi yang komprehensif dan representatif dari kode aplikasi. Oleh sebab itu, prinsip *clean code* yang diterapkan benar-benar dapat meningkatkan *maintainability* aplikasi Website monitoring sistem jurnal.

Peneliti mengambil sampel berdasarkan pengamatan dan keterlibatan langsung dengan code aplikasi. Peneliti akan berfokus pada evaluasi dan optimasi *maintainability* aplikasi dengan menerapkan prinsip *clean code*.

3.3 Kebutuhan Perangkat Penelitian

Dalam pelaksanaan penelitian ini, tentunya diperlukan perangkat yang mendukung keperluan penelitian. Kebutuhan perangkat penelitian dibagi menjadi dua, yaitu perangkat keras dan perangkat lunak. Deskripsi perangkat dan spesifikasi dijelaskan pada Tabel 3.1 untuk kebutuhan perangkat keras dan Tabel 3.2 untuk kebutuhan perangkat lunak.

3.3.1 Kebutuhan Perangkat Keras

Tabel 3.1 Kebutuhan Perangkat Keras

No.	Perangkat	Spesifikasi
1	<i>Device</i>	Laptop
2	<i>Processor</i>	AMD Ryzen 3 3250U
3	Memori (RAM)	12 Gb
4	Monitor	144Hz

3.3.2 Kebutuhan Perangkat Lunak

Tabel 3.2 Kebutuhan Perangkat Lunak

No.	Perangkat	Spesifikasi
1	Sistem Operasi	Windows 11
2	Bahasa Pemograman	Python dan php
3	<i>Word Processing</i>	Visual Studio Code
4	Peramban	Chrome

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Pengumpulan Data Smell Code

Dalam tahap awal, dilakukan analisis atau pengumpulan data *smell code* terhadap kode yang digunakan dalam aplikasi monitoring sistem jurnal. Aplikasi Website Monitoring Sistem jurnal memiliki beberapa file utama yang berguna dalam menjalankan sistem diantaranya beberapa file PHP (Halaman_home.php, Halaman_Jurnal.php, Halaman_Artikel.php, indeks.php) dan file Python (IndexScrapper.py dan JurnalScraper.py). Pengumpulan data ini dilakukan menggunakan *tools Sonarqube*, didapatkan 981 smell code issues adapun

Security 1 Open Issues		Reliability 54 Open Issues		Maintainability 981 Open Issues
Accepted issues 0 Valid issues that were not fixed		Coverage 0.0% On 2k lines to cover.		Duplications 3.5% On 22k lines.

hasil yang didapatkan sebagai berikut:

4.1.1 Masalah Awal dalam kode

a. Cycmatic Compaxity yang Tinggi

Beberapa fungsi memiliki terlalu banyak percabangan dan logika kompleks , yang menyulitkan pemeliharaan .

b. Pengulangan Kode

Terdapat banyak bagian kode yang diulang, meningkatkan risiko kesalahan dan membuat kode sulit untuk dipelihara dimasa depan .

c. Minimnya Dokumentasi dan Komentar.

Hal ini menyebabkan kesulitan bagi pengembang lain untuk memahami tujuan dan fungsi dari kode yang ditulis.

4.2 Pengukuran Maintainability Indeks Sebelum Refactorisasi

Berdasarkan analisis data yang ditemukan menunjukkan bahwa kode memiliki beberapa metrik penting yang mencerminkan kualitas dan kompleksitasnya. Matrik-matrik tersebut terdiri dari *line of Code* (LOC), *Cyclomatic Complexity* (CC), dan *Halstead Volume* (HV). Dilakukan pengolahan data pada masing masing file yang ada untuk menemukan matriks-marik tersebut.

4.2.1 Line of Code (LOC)

Jumlah baris kode dalam setiap file menunjukkan ukuran aplikasi dan dapat

mempengaruhi pemeliharaan serta pengembangan di masa depan.

Untuk menghitung nilai LOC, telah dibuat sebuah skrip Python yang secara otomatis menghitung jumlah baris kode dalam setiap file Python dan PHP.

Berikut adalah skrip untuk menghitung LOC pada file Py

```
def calculate_metrics(file_path):
    """Menghitung LOC, Cyclomatic Complexity, dan Halstead
    Volume"""
    code = read_file(file_path)
    if code is None:
        return None

    loc = count_loc_without_comments(code)

    # Menghitung kompleksitas siklomatis (Cyclomatic
    Complexity)
    cc_result = cc_visit(code)
    cc = sum(block.complexity for block in cc_result)

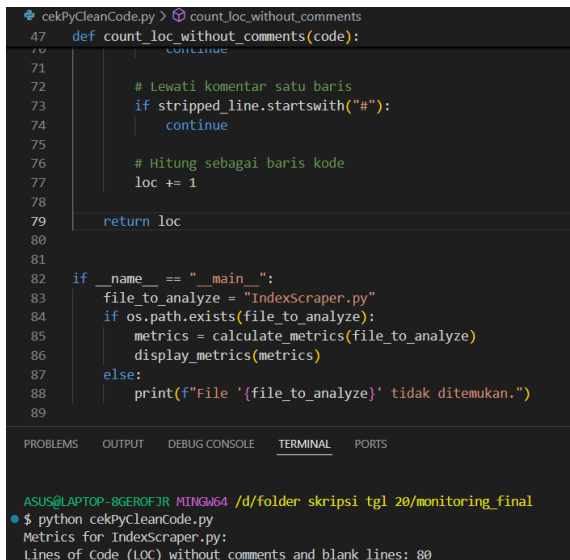
    # Menghitung Halstead Volume
    try:
        hv = h_visit(code).total.volume
    except Exception as e:
        print(f"Error calculating Halstead metrics: {e}")
        hv = 0

    return {"file": os.path.basename(file_path), "loc":
loc, "cc": cc, "hv": hv}

if __name__ == "__main__":
    file_to_analyze = "IndexScraper.py"
    if os.path.exists(file_to_analyze):
        metrics = calculate_metrics(file_to_analyze)
        display_metrics(metrics)
    else:
        print(f"File '{file_to_analyze}' tidak
ditemukan.")
```

Pada Skrip diatas dirancang agar tidak menghitung baris yang berisi komentar (diawali dan diakhiri dengan tanda #,""",atau "") serta baris kosong.

Hasil



```
cekPyCleanCode.py > count_loc_without_comments
47 def count_loc_without_comments(code):
48     continue
49
50     # Lewati komentar satu baris
51     if stripped_line.startswith("#"):
52         continue
53
54     # Hitung sebagai baris kode
55     loc += 1
56
57     return loc
58
59
60 if __name__ == "__main__":
61     file_to_analyze = "IndexScraper.py"
62     if os.path.exists(file_to_analyze):
63         metrics = calculate_metrics(file_to_analyze)
64         display_metrics(metrics)
65     else:
66         print(f"File '{file_to_analyze}' tidak ditemukan.")
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ASUS@LAPTOP-8GEROF3R MINGW64 /d/folder skripsi tgl 20/monitoring_final
$ python cekPyCleanCode.py
Metrics for IndexScraper.py:
Lines of Code (LOC) without comments and blank lines: 80
```

gambar 4. 1 Hasil Perhitungan LOC file Py Sebelum Refactoring

Berikut adalah skrip untuk menghitung LOC pada file PHP

```
def count_loc_without_comments(code):
    """Counts lines of code (LOC) excluding comments and
    blank lines."""
    lines = code.splitlines()
    loc = 0
    in_multiline_comment = False

    for line in lines:
        stripped_line = line.strip()

        if not stripped_line or
        stripped_line.startswith("//"):
            continue

        if stripped_line.startswith("/*"):
            in_multiline_comment = True

        if stripped_line.endswith("*/"):
            in_multiline_comment = False
            continue

        if in_multiline_comment:
            continue

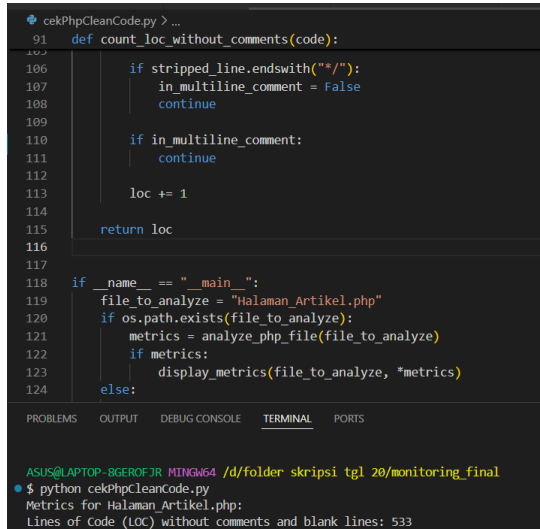
        loc += 1

    return loc

if __name__ == "__main__":
    file_to_analyze = "Halaman_Artikel.php"
    if os.path.exists(file_to_analyze):
        metrics = analyze_php_file(file_to_analyze)
        if metrics:
            display_metrics(file_to_analyze, *metrics)
    else:
        print(f"File '{file_to_analyze}' not found.")
```

Pada Skrip diatas dirancang agar tidak menghitung baris yang berisi komentar (diawali dengan tanda // atau /) serta baris kosong.

Hasil



The screenshot shows a code editor with a Python script named `cekPhpCleanCode.py`. The script defines a function `count_loc_without_comments` that iterates through lines of code, skipping comments and blank lines, and counts the remaining lines. The script then checks if it's running as the main program and analyzes a file named `Halaman_Artikel.php`. The terminal output shows the result of the analysis.

```
cekPhpCleanCode.py > ...
91 def count_loc_without_comments(code):
105
106     if stripped_line.endswith("*/"):
107         in_multiline_comment = False
108         continue
109
110     if in_multiline_comment:
111         continue
112
113     loc += 1
114
115     return loc
116
117
118 if __name__ == "__main__":
119     file_to_analyze = "Halaman_Artikel.php"
120     if os.path.exists(file_to_analyze):
121         metrics = analyze_php_file(file_to_analyze)
122         if metrics:
123             display_metrics(file_to_analyze, "metrics")
124     else:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ASUS@LAPTOP-BGEROFJR MINGW64 /d/folder skripsi tgl 20/monitoring_final
$ python cekPhpCleanCode.py
Metrics for Halaman_Artikel.php:
Lines of code (LOC) without comments and blank lines: 533
```

gambar 4. 2 Hasil Perhitungan LOC PHP Sebelum Refactoring

Skrip *full* akan dilampirkan dalam lampiran 1.
Berikut adalah hasil LOC sebelum *refactoring* berdasarkan prinsip *clean code* pada setiap file yang diuji dapat dilihat pada tabel :

Tabel 4. 1 LOC Sebelum Refactoring

File	LOC (Sebelum Refactoring)
IndexScraper.py	80
JurnalScraper.py	141
Halaman_Artikel.php	533
Indeks.php	163
Halaman_Home.php	163
Halaman_Jurnal.php	272
Total rata-rata mean	225.333

4.2.2 Cyclomatic Complexity (CC)

Cyclomatic Complexity mengukur jumlah arus independen melalui kode, yang mencerminkan kompleksitas logika. Rumus untuk menghitung Cyclomatic Complexity adalah sebagai berikut:

$$CC = E - N + 2P$$

Keterangan :

$CC = Cyclomatic\ complexity$

E = Jumlah edge (sisi) dalam graf kontrol alur

N = Jumlah node (simpul),

P = Jumlah komponen terhubung dalam graf

Untuk menghitung nilai CC, telah dibuat sebuah skrip Python yang secara otomatis menghitung jumlah baris kode dalam setiap file Python dan file PHP.

Berikut adalah Skrip untuk menghitung nilai CC pada file Python

```
import os
from radon.complexity import cc_visit
from radon.metrics import h_visit

def calculate_metrics(file_path):
    """Menghitung LOC, Cyclomatic Complexity, dan
    Halstead Volume"""
    code = read_file(file_path)
    if code is None:
        return None

    loc = count_loc_without_comments(code)

    # Menghitung kompleksitas siklomatis (Cyclomatic
    Complexity)
    cc_result = cc_visit(code)
    cc = sum(block.complexity for block in cc_result)

    # Menghitung Halstead Volume
    try:
        hv = h_visit(code).total.volume
    except Exception as e:
```

```

        print(f"Error calculating Halstead metrics: {e}")
        hv = 0

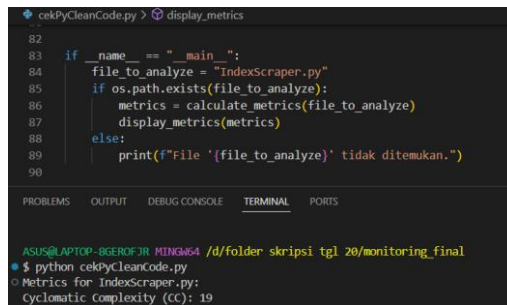
    return {"file": os.path.basename(file_path), "loc":
loc, "cc": cc, "hv": hv}

if __name__ == "__main__":
    file_to_analyze = "IndexScraper.py"
    if os.path.exists(file_to_analyze):
        metrics = calculate_metrics(file_to_analyze)
        display_metrics(metrics)
    else:
        print(f"File '{file_to_analyze}' tidak
ditemukan.")

```

Pada skrip ini memakai Library random sehingga kita hanya memanggil module random.complexity dan mengimport cc_visit untuk melakukan perhitungan nilai CC

Hasil



The screenshot shows a code editor with a Python script and its execution output in the terminal. The script is named 'cekPyCleanCode.py' and contains the following code:

```

82
83 if __name__ == "__main__":
84     file_to_analyze = "IndexScraper.py"
85     if os.path.exists(file_to_analyze):
86         metrics = calculate_metrics(file_to_analyze)
87         display_metrics(metrics)
88     else:
89         print(f"File '{file_to_analyze}' tidak ditemukan.")
90

```

The terminal output shows the command being executed and the resulting metrics:

```

ASUS@LAPTOP-BGEROF3R MINGW64 /d/folder skripsi tgl 20/monitoring_final
$ python cekPyCleanCode.py
Metrics for IndexScraper.py:
Cyclomatic Complexity (CC): 19

```

gambar 4. 3 Hasil Perhitungan CC File Py Sebelum Refactoring

Berikut adalah skrip untuk menghitung CC

Pada file PHP.

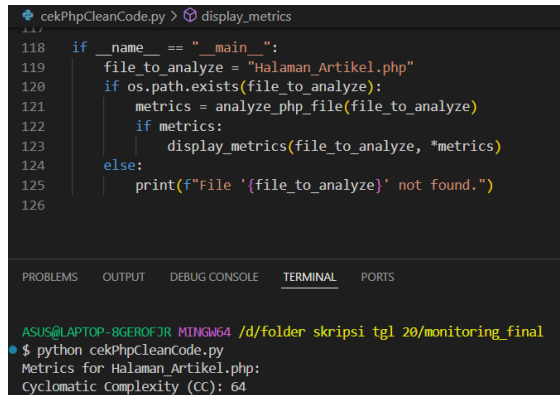
```
# Constants
INITIAL_CYCLOMATIC_COMPLEXITY = 1
PHP_OPERATORS_PATTERN = (
    r"[=+\-
    */<>!&|?:%]+|\b(if|else|for|foreach|while|switch|case|
    catch|return)\b"
)
PHP_OPERANDS_PATTERN = r"\b[a-zA-Z_][a-zA-Z0-9_]*\b"
CONTROL_FLOW_KEYWORDS = [
    "if",
    "else if",
    "for",
    "foreach",
    "while",
    "switch",
    "case",
    "catch",
]

def calculate_cyclomatic_complexity(code):
    """Calculates Cyclomatic Complexity (CC) based on
    control flow keywords."""
    cc = INITIAL_CYCLOMATIC_COMPLEXITY
    for keyword in CONTROL_FLOW_KEYWORDS:
        cc += len(re.findall(rf"\b{keyword}\b", code))
    return cc

if __name__ == "__main__":
    file_to_analyze = "Halaman_Artikel.php"
    if os.path.exists(file_to_analyze):
        metrics = analyze_php_file(file_to_analyze)
        if metrics:
            display_metrics(file_to_analyze, *metrics)
    else:
        print(f"File '{file_to_analyze}' not found.")
```

Pada skrip ini function `calculate_cyclomatic_complexity` berfungsi mencari kata kunci kontrol alur dalam kode php dan menjumlah kata kunci kontrol alur seperti if, for, while, switch, and case untuk menentukan nilai CC

Hasil



```
cekPhpCleanCode.py > display_metrics
118 if __name__ == "__main__":
119     file_to_analyze = "Halaman_Artikel.php"
120     if os.path.exists(file_to_analyze):
121         metrics = analyze_php_file(file_to_analyze)
122         if metrics:
123             display_metrics(file_to_analyze, *metrics)
124     else:
125         print(f"File '{file_to_analyze}' not found.")
126
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ASUS@LAPTOP-8GEROFJR MINGW64 /d/folder skripsi tgl 20/monitoring_final
$ python cekPhpCleanCode.py
Metrics for Halaman_Artikel.php:
Cyclomatic Complexity (CC): 64
```

gambar 4. 4 Hasil Perhitungan CC file PHP Sebelum Refactoring

Skrip *full* akan dilampirkan dalam lampiran 2.

Berikut adalah nilai CC sebelum dilakukannya refactoring sesuai dengan prinsip clean code pada masing-masing file.

Tabel 4. 2 CC Sebelum Refactoring

File	CC (Sebelum Refactoring)
IndexScraper.py	19
JurnalScraper.py	31
Halaman_Artikel.php	64
Indeks.php	8
Halaman_Home.php	8
Halaman_Jurnal.php	23
Total rata-rata	25.50

4.2.3 Helstead Volume (HV)

Halstead Volume mengukur ukuran logis dari kode berdasarkan jumlah operator dan operand. Rumus menghitung *Halstead Volume* sebagai berikut :

$$HV = N \times \log_2(\eta)$$

$$N = N1 + N1 \text{ (total operator dan operand)}$$

$$\eta = \eta_1 + \eta_2 \text{ (jumlah unik operator dan operand)}$$

Untuk menghitung nilai HV, telah dibuat sebuah skrip Python yang secara otomatis menghitung jumlah baris kode dalam setiap file Python dan File php.

Berikut adalah Skrip untuk menghitung nilai HV pada file py.

```
import os
from radon.complexity import cc_visit
from radon.metrics import h_visit

def calculate_metrics(file_path):
    """Menghitung LOC, Cyclomatic Complexity, dan
    Halstead Volume"""
    code = read_file(file_path)
    if code is None:
        return None

    loc = count_loc_without_comments(code)

    # Menghitung kompleksitas siklomatis (Cyclomatic
    Complexity)
    cc_result = cc_visit(code)
    cc = sum(block.complexity for block in cc_result)

    # Menghitung Halstead Volume
    try:
        hv = h_visit(code).total.volume
    except Exception as e:
        print(f"Error calculating Halstead metrics:
        {e}")
        hv = 0

    return {"file": os.path.basename(file_path), "loc":
    loc, "cc": cc, "hv": hv}
```

```

if __name__ == "__main__":
    file_to_analyze = "IndexScraper.py"
    if os.path.exists(file_to_analyze):
        metrics = calculate_metrics(file_to_analyze)
        display_metrics(metrics)
    else:
        print(f"File '{file_to_analyze}' tidak
ditemukan.")

```

kita hanya memanggil module random.metrics dan mengimport h_visit untuk melakukan perhitungan nilai HV

Hasil

The screenshot shows a code editor with the following Python code in `IndexScraper.py`:

```

79 | print(f"Cyclomatic Complexity (CC): {metrics['cc']}")
80 |
81 |
82 | if __name__ == "__main__":
83 |     file_to_analyze = "IndexScraper.py"
84 |     if os.path.exists(file_to_analyze):
85 |         metrics = calculate_metrics(file_to_analyze)
86 |         display_metrics(metrics)
87 |     else:
88 |         print(f"File '{file_to_analyze}' tidak ditemukan.")
89 |

```

Below the code editor, the terminal output shows the command and result:

```

ASUS@LAPTOP-BGEROFJR MINGW64 /d/folder skripsi tgl 20/monitoring_final
$ python cekPyCleanCode.py
Metrics for IndexScraper.py:
Halstead Volume (HV): 138.3016990363956

```

gambar 4. 5 Hasil Perhitungan HV file Py Sebelum Refactoring

**Berikut adalah Skrip untuk menghitung nilai HV
pada file PHP.**

```
# Constants
INITIAL_CYCLOMATIC_COMPLEXITY = 1
PHP_OPERATORS_PATTERN = (
    r"[=+\-
    */<>!&|?:%]+\b(if|else|for|foreach|while|switch|case|catch|ret
    urn)\b"
)
PHP_OPERANDS_PATTERN = r"\b[a-zA-Z_][a-zA-Z0-9_]*\b"
CONTROL_FLOW_KEYWORDS = [
    "if",
    "else if",
    "for",
    "foreach",
    "while",
    "switch",
    "case",
    "catch",
]

def extract_operators_operands(code):
    """Extracts operators and operands from PHP code."""
    operators = re.findall(PHP_OPERATORS_PATTERN, code)
    operands = re.findall(PHP_OPERANDS_PATTERN, code)
    return operators, operands

def calculate_halstead_volume(code):
    """Calculates Halstead Volume (HV) based on operators and
    operands."""
    operators, operands = extract_operators_operands(code)

    unique_operators = set(operators)
    unique_operands = set(operands)

    total_operators = len(operators)
    total_operands = len(operands)

    n1 = len(unique_operators)
    n2 = len(unique_operands)
    N1 = total_operators
    N2 = total_operands

    n = n1 + n2
    N = N1 + N2
```

```

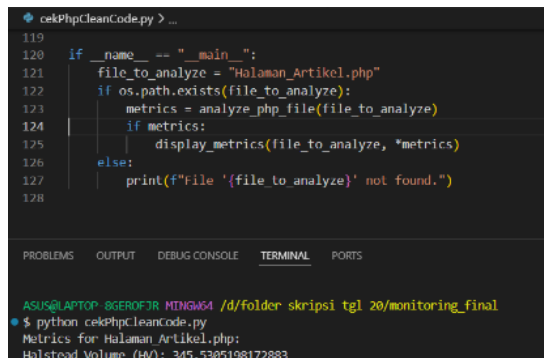
        adjustment_factor = 0.01
        return N * (math.log2(n) * adjustment_factor if n > 1
        else 0) if n > 0 else 0

if __name__ == "__main__":
    file_to_analyze = "Halaman_Artikel.php"
    if os.path.exists(file_to_analyze):
        metrics = analyze_php_file(file_to_analyze)
        if metrics:
            display_metrics(file_to_analyze, *metrics)
    else:
        print(f"File '{file_to_analyze}' not found.")

```

Pada skrip ini, function calculate_halstead_volume merupakan function main untuk menghitung HV ,pertama- tama memanggil extract_operators_operands untuk menghitung jumlah operator unik (n1), operand unik (n2) , total operator (N1), dan total operant (N2) kemudian nilai HV dihitung dengan rumus HV.

Hasil



```

cekPhpCleanCode.py > ...
119
120 if __name__ == "__main__":
121     file_to_analyze = "Halaman_Artikel.php"
122     if os.path.exists(file_to_analyze):
123         metrics = analyze_php_file(file_to_analyze)
124         if metrics:
125             display_metrics(file_to_analyze, *metrics)
126     else:
127         print(f"File '{file_to_analyze}' not found.")
128
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

ASUS@LAPTOP-B6EROF3R MINGW64 /d/folder skripsi tgl 20/monitoring_final
* $ python cekPhpCleanCode.py
Metrics for Halaman_Artikel.php:
Halstead Volume (HV): 345.3365198172883

```

gambar 4. 6 Hasil Perhitungan HV file PHP Sebelum Refactoring

Skrip *full* akan dilampirkan dalam lampiran 2.

Berikut adalah nilai HV sebelum dilakukannya refactoring sesuai dengan prinsip clean code pada masing-masing file.

Tabel 4. 3 HV Sebelum Refactoring

File	HV (Sebelum Refactoring)
IndexScraper.py	138.30
JurnalScraper.py	99.40
Halaman_Artikel.php	345.50
Indeks.php	157.11
Halaman_Home.php	157.11
Halaman_Jurnal.php	216.84
Total rata-rata	185.71

4.2.4 Nilai *Maintainability Indeks* (MI)

Adapun rumus maintainability indeks sebagai berikut:

$$MI = 171 - 5.2 \times \ln(LOC) - 0.23 \times CC - 16.2 \times \ln(HV)$$

Keterangan :

MI: *Maintainability Index* (Indeks Kemampuan Pemeliharaan)

LOC: *Lines of Code* (Jumlah Baris Kode)

CC: *Cyclomatic Complexity* (Kompleksitas Siklomatik)

HV: *Volume Halstead*

ln: Logaritma natural

Untuk menghitung nilai MI, telah dibuat sebuah skrip Python yang secara otomatis menghitung jumlah baris kode dalam setiap file Python dan File php.

Berikut adalah Skrip untuk menghitung nilai MI file IndexScraper.py.

```
import math

def calculate_mi(cc, hv, loc):
    """
    Menghitung Maintainability Index (MI)
    berdasarkan formula standar.
    Args:
        cc (int): Cyclomatic Complexity
        hv (float): Halstead Volume
        loc (int): Lines of Code
    Returns:
        float: Nilai Maintainability Index (MI)
    """
    try:
        mi = 171 - 5.2 * math.log(loc) - 0.23 * cc
        - 16.2 * math.log(hv)
        return round(mi, 2)
    except ValueError as e:
        return f"Error: {e}. Pastikan nilai LOC
        dan HV lebih besar dari nol."

# Contoh penggunaan
cc = 20 # Cyclomatic Complexity
hv = 98.09 # Halstead Volume
loc = 69 # Lines of Code

mi = calculate_mi(cc, hv, loc)
print(f"Maintainability Index (MI): {mi}")
```

Pada skrip ini, perhitungan MI dilakukan dengan memanggil function `calculate_mi`, yang menggunakan nilai CC, HV, dan LOC sebagai input. Kemudian skrip ini akan menghitung nilai yang sudah di input dengan rumus MI.

Hasil

```
cekMICleanCode.py > ...
1 import math
2
3
4 def calculate_mi(cc, hv, loc):
5     """
6     Menghitung Maintainability Index (MI) berdasarkan formula standar.
7     Args:
8         cc (int): Cyclomatic Complexity
9         hv (float): Halstead Volume
10        loc (int): Lines of Code
11    Returns:
12        float: Nilai Maintainability Index (MI)
13    """
14    try:
15        mi = 171 - 5.2 * math.log(loc) - 0.23 * cc - 16.2 * math.log(hv)
16        return round(mi, 2)
17    except ValueError as e:
18        return f"Error: {e}. Pastikan nilai LOC dan HV lebih besar dari nol."
19
20
21 # Contoh penggunaan
22 cc = 19 # Cyclomatic Complexity
23 hv = 138.30 # Halstead Volume
24 loc = 80 # Lines of Code
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

ASUS@LAPTOP-BGEROFJR MINGW64 /c/xampp/htdocs/codeInaRefactoring[GL28
\$ python cekMICleanCode.py
Maintainability Index (MI): 63.99

gambar 4. 7 Hasil Perhitungan MI Sebelum Refactoring

Berikut adalah nilai MI sebelum dilakukannya refactoring sesuai dengan prinsip clean code pada masing-masing file.

Tabel 4. 4 MI Sebelum Refactoring

File	MI (Sebelum Refactoring)
IndexScraper.py	63.99
JurnalScraper.py	63.7
Halaman_Artikel.php	28.75

Indeks.php	60.79
Halaman_Home.php	60.79
Halaman_Jurnal.php	49.42
Total rata-rata	54.71

4.3 Melakukan Refactoring (penerapan Prinsip *Clean Code*)

Proses refactoring berfokus pada penerapan prinsip Clean Code untuk meningkatkan keterbacaan, modularitas dan *maintainability* kode. Prinsip ini digunakan untuk memecah fungsi yang besar dan kompleks menjadi lebih modular , serta mengurangi duplikasi kode. Proses ini dilakukan berdasarkan hasil dari tahapan pengumpulan data smell code. Adapun hasilnya refactoring sebagai berikut :

4.3.1 Halaman Artikel.php

Sebelum refactoring sebagaimana pada lampiran 3, kode masih mengandung elemen HTML yang tidak perlu dan pemanggilan element berulang dalam satu file. Sehingga struktur Kode kurang modular karena mencampurkan elemen

tampilan dan pemrosesan data dalam satu file. Hal ini menyebabkan masalah *Shotgun Surgery* dan *Duplication Code* dimana perubahan di satu bagian kode menyebabkan banyak perubahan di tempat lain karena elemen yang tidak modular. Oleh karena itu pada kode sesudah refactoring (lampiran 4) diterapkan nya Single Responsibility Principle (SRP) dimana setiap bagian kode hanya memiliki satu tanggung jawab dengan memisahkan tampilan dan logika pemrosesan data. Selain itu skrip javascript dalam file sebelum refactoring terlalu bernested ,sehingga pada file sesudah skrip javascript dibuat lebih modular.

Kode sebelum

```
<?php
    include "Koneksi.php";
?>

<!doctype html>
<!--[if lt IE 7]>    <html class="no-js lt-ie9 lt-ie8 lt-ie7"
lang=""> <![endif-->
<!--[if IE 7]>    <html class="no-js lt-ie9 lt-ie8"
lang=""> <![endif-->
<!--[if IE 8]>    <html class="no-js lt-ie9" lang="">
<![endif-->
<!--[if gt IE 8]><!--> <html class="no-js" lang=""> <!--
<![endif-->
<head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
    <!-- <meta http-equiv="refresh" content="180"> -->

    <title>Halaman Artikel</title>
    <meta name="description" content="Monitoring Jurnal">
    <meta name="viewport" content="width=device-width, initial-
scale=1">

    <link rel="apple-touch-icon" href="images/uinws.png">
    <link rel="shortcut icon" href="images/uinws.png">

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/normalize.css@8.0.0/normalize
.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/boot
strap.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/font-awesome@4.7.0/css/font-
awesome.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/gh/lykmapipo/themify-
icons@0.1.2/css/themify-icons.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/pixeden-stroke-7-
icon@1.2.3/pe-icon-7-stroke/dist/pe-icon-7-stroke.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/flag-icon-
css/3.2.0/css/flag-icon.min.css">
    <link rel="stylesheet" href="assets/css/cs-skin-
elastic.css">
    <link rel="stylesheet" href="assets/css/style.css">
</script>
```

Tercampurnya elemen tampilan dan pemrosesan data dalam satu file sehingga dapat menyebabkan *Shotgun Surgery*.

```
document
    .getElementById("startYearFilter")
    .addEventListener("change", function () {
        var selectedStartYear = this.value;
        var selectedEndYear =
document.getElementById("endYearFilter").value;
        var selectedJurnal =
document.getElementById("jurnalFilter").value;
        var keyword =
document.getElementById("keywordSearch").value;
        var params = new
URLSearchParams(window.location.search);
        if (selectedStartYear !== "none") {
            params.set("start_year", selectedStartYear);
        } else {
            params.delete("start_year");
        }
        if (selectedEndYear !== "none") {
            params.set("end_year", selectedEndYear);
        } else {
            params.delete("end_year");
        }
        if (selectedJurnal !== "none") {
            params.set("jurnal", selectedJurnal);
        } else {
            params.delete("jurnal");
        }
        if (keyword !== "") {
            params.set("keyword", keyword);
        } else {
            params.delete("keyword");
        }
        window.location.href = "Halaman_Artikel.php?" +
params.toString();
    });

document
    .getElementById("endYearFilter")
    .addEventListener("change", function () {
        var selectedStartYear =
document.getElementById("startYearFilter").value;
        var selectedEndYear = this.value;
        var selectedJurnal =
document.getElementById("jurnalFilter").value;
```

Kode terlalu bernested, Sehingga kode sulit dibaca dan dipahami ,meningkatkan nilai CC dan menyulitkan debugging dan testing.

Kode Sesudah.

```
<?php
include "Koneksi.php";
?>

<body>
    <?php include "left_panel.php"?>
    <!-- /#left-panel -->
    <!-- Right Panel -->
    <div id="right-panel" class="right-panel">
        <!-- Header-->
        <?php include "header.php"?>
        <!-- /#header -->
        <?php include "footer.php"?>
        <script src="./assets/js/scriptArtikel.js"></script>
    </body>
```

Pemisahan kode untuk tampilan dan logika, sehingga menghindari *Shotgun Surgery*.

```

function updateURLParams() {
    const params = new
URLSearchParams(window.location.search);
    const filters = {
        start_year:
document.getElementById("startYearFilter").value,
        end_year:
document.getElementById("endYearFilter").value,
        jurnal: document.getElementById("jurnalFilter").value,
        keyword: document.getElementById("keywordSearch").value,
        sort_by: document.getElementById("sortFilter")?.value,
        page: document.getElementById("jump_page")?.value,
    };

    Object.entries(filters).forEach(([key, value]) => {
        if (value && value !== "none") {
            params.set(key, value);
        } else {
            params.delete(key);
        }
    });

    window.location.href = "Halaman_Artikel.php?" +
params.toString();
}

["startYearFilter", "endYearFilter", "jurnalFilter",
"sortFilter"].forEach(
    (id) => {
        document.getElementById(id)?.addEventListener("change",
updateURLParams);
    }
);

document
    .getElementById("keywordSearch")
    ?.addEventListener("keypress", function (e) {
        if (e.key === "Enter") {
            e.preventDefault();
            updateURLParams();
        }
    });

document
    .getElementById("jump_page_form")
    ?.addEventListener("submit", function (e) {
        e.preventDefault();
        updateURLParams();
    });

document
    .getElementById("clear_filters")
    ?.addEventListener("click", function () {

```

Mengurangi nested kode pada Skrip javascript Dengan cara membuat funtion utama sehingga semua filter diperbarui dalam satu fungsi, sehingga mengurangi pengulangan kode (DRY-Don't Repeat Yourself). Selain itu dengan dibuatnya object filters memudahkan mengembangkan jika ada filters baru yang ingin ditambah ,dengan pendekatan ini perubahan tidak menyebar ke banyak tempat (menghindari *Shotgun Surgery*)

4.3.2 Indeks.php

Sebelum dilakukannya refactoring sebagaimana terdapat pada lampiran 5, kode masih mengandung element HTML lama dan banyak pemanggilan eksternal yang tidak diperlukan. Struktur kode kurang optimal karena terlalu banyak dependensi yang bisa dioptimalkan

Kode Sebelum refactoring masih mengandung elemen HTML yang tidak perlu dan pemanggilan element berulang dalam satu file. Sehingga struktur Kode kurang modular karena mencampurkan elemen tampilan dan pemrosesan data dalam satu file. Hal ini

menyebabkan masalah *Shotgun Surgery* dan *Duplication Code* dimana perubahan di satu bagian kode menyebabkan banyak perubahan di tempat lain karena elemen yang tidak modular. Oleh karena itu pada kode sesudah refactoring (lampiran 6) diterapkan nya Single Responsability Principle (SRP) dimana setiap bagian kode hanya memiliki satu tanggung jawab dengan memisahkan tampilan dan logika pemrosesan data.

Sebelum Refactoring

```
<?php
<!doctype html>
<!--[if lt IE 7]>          <html class="no-js lt-ie9 lt-ie8 lt-
ie7" lang=""> <![endif]-->
<!--[if IE 7]>            <html class="no-js lt-ie9 lt-ie8"
lang=""> <![endif]-->
<!--[if IE 8]>            <html class="no-js lt-ie9" lang="">
<![endif]-->
<!--[if gt IE 8]><!-->
<html class="no-js" lang=""> <!--<![endif]-->

<head>
  <meta charset="utf-8">
  <!-- <meta http-equiv="refresh" content="180"> -->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Halaman Home</title>
  <meta name="description" content="Monitoring Jurnal">
  <meta name="viewport" content="width=device-width,
initial-scale=1">

  <link rel="apple-touch-icon" href="images/uinws.png">
  <link rel="shortcut icon" href="images/uinws.png">

  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/normalize.css@8.0.0/norm
```

```

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/normalize.css@8.0.0/normali
ze.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bo
otstrap.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/font-
awesome@4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/gh/lykmapipo/themify-
icons@0.1.2/css/themify-icons.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/pixeden-stroke-7-
icon@1.2.3/pe-icon-7-stroke/dist/pe-icon-7-stroke.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/flag-icon-
css/3.2.0/css/flag-icon.min.css">
    <link rel="stylesheet" href="assets/css/cs-skin-
elastic.css">
    <link rel="stylesheet" href="assets/css/style.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery
.min.js"></script>
    <!-- <script type="text/javascript"
src="https://cdn.jsdelivr.net/html5shiv/3.7.3/html5shiv.min.js
"></script> -->
    <link
href="https://cdn.jsdelivr.net/npm/chartist@0.11.0/dist/charti
st.min.css" rel="stylesheet">
    <link
href="https://cdn.jsdelivr.net/npm/jqvmap@1.5.1/dist/jqvmap.mi
n.css" rel="stylesheet">

    <link
href="https://cdn.jsdelivr.net/npm/weathericons@2.1.0/css/weat
her-icons.css" rel="stylesheet" />
    <link
href="https://cdn.jsdelivr.net/npm/fullcalendar@3.9.0/dist/ful
lcalendar.min.css" rel="stylesheet" />

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/1.5.3/jspdf.
min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf-
autotable/3.5.6/jspdf.plugin.autotable.min.js"></script>
    <script type="text/javascript"
src="assets/js/pdf/main.js"></script>

</head>
<body>
    <!-- Left Panel -->

```

```

<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Halaman home</title>
  <meta name="description" content="Monitoring Jurnal">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
</head>

<body>
  <?php include "left_panel.php"?>
  <div id="right-panel" class="right-panel">
    <?php include "header.php"?>
    <?php include "footer.php"?>
  </div>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/
bootstrap.min.js"></script>
</body>

```

Pemisahan kode untuk tampilan dan logika, sehingga menghindari *Shotgun Surgery*. Full code ada pada lampiran 6

4.3.3 Halaman_Jurnal.php

Pada file ini sebelum refactoring (lampiran 7) dilakukan refactoring bertujuan untuk meningkatkan keamanan, mengurangi pengulangan code, serta membuat code lebih modular dengan memisahkan logika pengolahan data jurnal dari tampilan HTML. Sehingga menyebabkan *Shotgun Surgery* dan *Duplication*

Code dimana perubahan di satu bagian kode menyebabkan banyak perubahan di tempat lain karena elemen yang tidak modular. Oleh karena itu pada kode sesudah refactoring diterapkan nya Single Responsobility Principle (SRP) dimana setiap bagian kode hanya memiliki satu tanggung jawab dengan memisahkan tampilan dan logika pemrosesan data. Selain itu skrip javascript dibuat lebih modular agar skrip mudah dipahami.

Sebelum Refaktorisasi

```
<!doctype html>
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8 lt-ie7"
lang=""> <![endif]-->
<!--[if IE 7]>         <html class="no-js lt-ie9 lt-ie8"
lang=""> <![endif]-->
<!--[if IE 8]>         <html class="no-js lt-ie9" lang="">
<![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js" lang=""> <!--
<![endif]-->
<head>
  <meta charset="utf-8">
  <!-- <meta http-equiv="refresh" content="180"> -->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Halaman Jurnal</title>
  <meta name="description" content="Monitoring Jurnal">
  <meta name="viewport" content="width=device-width, initial-
scale=1">

  <link rel="apple-touch-icon" href="images/uinws.png">
  <link rel="shortcut icon" href="images/uinws.png">

  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/normalize.css@8.0.0/normaliz
e.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/bootstrap@4.1.3/dist/css/boo
tstrap.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/font-awesome@4.7.0/css/font-
```

```

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/gh/lykmapipo/themify-
icons@0.1.2/css/themify-icons.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/pixeden-stroke-7-
icon@1.2.3/pe-icon-7-stroke/dist/pe-icon-7-stroke.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/flag-icon-
css/3.2.0/css/flag-icon.min.css">
    <link rel="stylesheet" href="assets/css/cs-skin-
elastic.css">
    <link rel="stylesheet" href="assets/css/style.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jque
ry.min.js"></script>
    <!-- <script type="text/javascript"
src="https://cdn.jsdelivr.net/html5shiv/3.7.3/html5shiv.min.
js"></script> -->
    <link
href="https://cdn.jsdelivr.net/npm/chartist@0.11.0/dist/char
tist.min.css" rel="stylesheet">
    <link
href="https://cdn.jsdelivr.net/npm/jqvmap@1.5.1/dist/jqvmap.
min.css" rel="stylesheet">

    <link
href="https://cdn.jsdelivr.net/npm/weathericons@2.1.0/css/we
ather-icons.css" rel="stylesheet" />
    <link
href="https://cdn.jsdelivr.net/npm/fullcalendar@3.9.0/dist/f
ullcalendar.min.css" rel="stylesheet" />

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/1.5.3/jspd
f.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf-
autotable/3.5.6/jspdf.plugin.autotable.min.js"></script>
    <script type="text/javascript"
src="assets/js/pdf/main.js"></script>

</head>

<body>
    <!-- Left Panel -->
    <aside id="left-panel" class="left-panel">
        <nav class="navbar navbar-expand-sm navbar-default">
            <div id="main-menu" class="main-menu collapse
navbar-collapse">
                <ul class="nav navbar-nav">
                    <li class="">
                        <a href="Halaman_Home.php"><i
class="menu-icon fa fa-home"></i>Halaman Home </a>

```

Tercampurnya elemen tampilan dan pemrosesan data dalam satu file sehingga dapat menyebabkan *Shotgun Surgery*.

```
document
  .getElementById("sortingFilter")
  .addEventListener("change", function () {
    updateQueryParams();
  });

document
  .getElementById("keywordSearch")
  .addEventListener("keydown", function (e) {
    if (e.key === "Enter") {
      e.preventDefault();
      updateQueryParams();
    }
  });

function updateQueryParams() {
  const sorting =
document.getElementById("sortingFilter").value;
  const keyword =
document.getElementById("keywordSearch").value;
  let queryParams = "?page=1";

  if (sorting !== "none") {
    queryParams += "&sorting=" + sorting;
  }
  if (keyword) {
    queryParams += "&keyword=" +
encodeURIComponent(keyword);
  }

  window.location.href = window.location.pathname
+ queryParams;
}
```

Pada kode ini langsung menambahkan event listener tanpa pemisahan fungsi, yang bisa membuat debugging lebih sulit dan jika ada perubahan harus langsung dilakukan di event listener utama, berisiko menyebabkan *Shotgun Surgery*.

Sesudah Refactorisasi.

```
<!doctype html>
<html class="no-js" lang="">

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <title>Halaman Jurnal</title>
  <meta name="description" content="Monitoring
Jurnal">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
</head>

<body>
  <!-- Left Panel -->
  <?php include "left_panel.php"?>
  <!-- /#left-panel -->
  <!-- Right Panel -->
  <div id="right-panel" class="right-panel">
    <!-- Header-->
    <?php include "header.php"?>
    <!-- /#header -->
    <?php include "footer.php"?>
    <!-- /.site-footer -->
  </div>
  <!-- /#right-panel -->
  <script src="./assets/js/scriptJurnal.js"></script>
</body>
```

Pemisahan kode untuk tampilan dan logika, sehingga menghindari *Shotgun Surgery*.

```
document
.getElementById("sortingFilter")
.addEventListner("change", function () {
    updateQueryParams();
});
document
.getElementById("keywordSearch")
.addEventListner("keydown", function (e) {
    if (e.key === "Enter") {
        e.preventDefault();
        updateQueryParams();
    }
});

function updateQueryParams() {
    const sorting =
document.getElementById("sortingFilter").value;
    const keyword =
document.getElementById("keywordSearch").value;
    let queryParams = "?page=1";

    if (sorting !== "none") {
        queryParams += "&sorting=" + sorting;
    }
    if (keyword) {
        queryParams += "&keyword=" +
encodeURIComponent(keyword);
    }

    window.location.href = window.location.pathname +
queryParams;
}
```

Pada kode ini memisahkan tanggung jawab antara event listener dan logika query parameter,

sehingga jika ingin menambah fitur baru, cukup ubah fungsi spesifik (event listener) tanpa mengubah logika utama. Full kode setelah refactoring terdapat dilampiran 8

4.3.4 Halaman_Home.php

Pada file sebelum refactoring (lampiran 9) fungsi `runUpdateScript` diciptakan untuk memisahkan logika bisnis dari tampilan, dan mengurangi dupikasi code code code dalam memanggil proses update. Serta Terdapat *Duplication code* dan *Shotgun Surgery*.

Sebelum refactoring

```
<!doctype html>
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8
lt-ie7" lang=""> <![endif]-->
<!--[if IE 7]>        <html class="no-js lt-ie9 lt-
ie8" lang=""> <![endif]-->
<!--[if IE 8]>        <html class="no-js lt-ie9"
lang=""> <![endif]-->
<!--[if gt IE 8]><!-->
<html class="no-js" lang=""> <!--<![endif]-->

<head>
  <meta charset="utf-8">
  <!-- <meta http-equiv="refresh" content="180"> -->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Halaman Home</title>
  <meta name="description" content="Monitoring Jurnal">
  <meta name="viewport" content="width=device-width,
initial-scale=1">

  <link rel="apple-touch-icon" href="images/uinws.png">
  <link rel="shortcut icon" href="images/uinws.png">
```

Tercampurnya elemen tampilan dan pemrosesan data dalam satu file sehingga dapat menyebabkan *Shotgun Surgery*.

Setelah Refaktorisasi

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <title>Halaman home</title>
  <meta name="description" content="Monitoring
Jurnal">
  <meta name="viewport" content="width=device-
width, initial-scale=1">
</head>

<body>
  <?php include "left_panel.php"?>
  <div id="right-panel" class="right-panel">
    <?php include "header.php"?>
  <div class="clearfix"></div>
    <?php include "footer.php"; ?>
  </div>
</body>
```

Pemisahan antara antarmuka dan logika menghilangkan code smell *Shotgun Surgery*. Full kode setelah refactoring terdapat pada lampiran 10

4.3.5 IndexScraper.py

Pada file sebelum refactoring (lampiran 11), skrip kode sebelum refactoring. Masih memiliki struktur yang kurang modular, dengan fungsi yang mencampur proses scraping dan database dalam satu blok kode. Hal ini menyebabkan terjadinya Long Function Dimana fungsi terlalu Panjang dan mencampurkan banyak tugas. Selain itu terdapat beberapa Duplicate Kode, serta manajemen koneksi database yang masih dilakukan secara manual. Oleh karena itu pada kode setelah refactoring di terapkanya Single Responsibility Principles (SRP) untuk mengatasi Long Function, Dimana kode dijadikan lebih modular.

Sebelum Refactoring

```
import requests
from bs4 import BeautifulSoup
import mysql.connector
import datetime
import time # for rate limiting (optional)

def scrape_journals_uin_walisongo(search_url,
indeks_jurnal_id):
    """
    Scrapes the accreditation of journals listed on the
    SINTA search page
    that are affiliated with UIN Walisongo Semarang.

    Args:
        search_url (str): The URL of the search page on
        SINTA website.
        indeks_jurnal_id (int): The ID of the journal in
        the database.

    Returns:
        str: The extracted accreditation (or None if not
        found).
    """

    response = requests.get(search_url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Find all journal listings
    journal_listings = soup.find_all('div', class_='list-
item row mt-3')

    journals = []
    for listing in journal_listings:
        # Check affiliation location (more generic
        selector)
        affiliation_element = listing.find('div',
class=lambda class_name: class_name and 'affil-loc' in
class_name and 'mt-2' in class_name)
        if affiliation_element and "Walisongo" in
        affiliation_element.text:
            # Extract journal name
            journal_name_element = listing.find('div',
```

Function terlalu Panjang dan melakukan terlalu banyak tugas. Hal ini akan menyebabkan terjadinya *Shotgun Surgery*.

```
db = mysql.connector.connect(  
    host="localhost",  
    username="monito29_admin",  
    password="monito29_admin",  
    database="monito29_db",  
    charset="utf8"  
)
```

Koneksi database dibuat menggunakan metode manual hal ini menyebabkan potensi Resource Leak, kebocoran resource jika koneksi tidak ditutup dengan benar.

```
response = requests.get(search_url)  
soup = BeautifulSoup(response.content,  
    'html.parser')
```

Pada kode ini fetch data dari web langsung didalam fungsi utama, hal ini menyebabkan adanya Duplicate Code pada setiap fungsi dan pada kode ini tidak ada error handling sehingga jika request gagal, program tetap mencoba parsing halaman error, yang bisa menyebabkan bug.

Sesudah Refaktorisasi

```
import requests
from bs4 import BeautifulSoup
import mysql.connector

def fetch_page_content(url):
    """Fetches and returns the HTML content of a given
    URL."""
    response = requests.get(url)
    response.raise_for_status()
    return BeautifulSoup(response.content, "html.parser")

def scrape_journal_accreditation(search_url):
    """Scrapes accreditation data for journals affiliated
    with UIN Walisongo from SINTA."""
    soup = fetch_page_content(search_url)
    journals = []

    for listing in soup.find_all("div", class_="list-item
row mt-3"):
        affiliation = listing.find("div", class_=lambda
x: x and "affil-loc" in x)
        if affiliation and "Walisongo" in
affiliation.text:
            journal_name = listing.find("div",
class_="affil-name mb-3")
            journal_name =
journal_name.find("a").text.strip() if journal_name else
None

            accreditation = listing.find(
                "span", class_=lambda x: x and
"accredited" in x
            )
            accreditation = (
                accreditation.text.strip().split()[0] if
accreditation else None
            )

            journals.append({"name": journal_name,
"accreditation": accreditation})

    return journals
```

```

def scrape_sinta_index(search_url):
    """Scrapes the Sinta index value from a journal
    profile page."""
    soup = fetch_page_content(search_url)
    sinta_card = soup.find("div", class_="card bg-card-
orange")

    if sinta_card:
        sinta_value = sinta_card.find("div",
class_="stat-num").text.split()[1]
        return f"S{sinta_value}"
    return None

def update_journal_accreditation(cursor, journal_id,
accreditation):
    """Updates the journal accreditation in the
    database."""
    cursor.execute(
        "UPDATE jurnal SET indeks_jurnal = %s WHERE
id_jurnal = %s",
        (accreditation, journal_id),
    )

def process_journals(cursor, journals):
    """Processes journals and updates their
    accreditation in the database."""
    for idx, (journal_id, url) in enumerate(journals,
start=1):
        print(f"Processing journal {idx}: ", end="")

        journals_data =
scrape_journal_accreditation(url)
        if journals_data:
            for journal in journals_data:
                if journal["accreditation"]:
                    update_journal_accreditation(
                        cursor, journal_id,
journal["accreditation"])
                print(
                    f"Accreditation:
{journals_data[journal['accreditation']]}, ID: {journal_id}"
                )

```

perubahan utama meliputi pemisahan logika pengambilan data jurnal, ekstraksi informasi, dan penyimpanan data ke dalam fungsi-fungsi kecil yang modular.

```
with mysql.connector.connect(**db_config) as db:
```

Menggunakan With untuk memastikan koneksi data base ditutup secara otomatis menghindari potensi terjadinya Resource Leak.

```
def fetch_page_content(url):  
    """Fetches and returns the HTML content of a given  
    URL."""  
    response = requests.get(url)  
    response.raise_for_status()  
    return BeautifulSoup(response.content,  
        "html.parser")
```

Pada kode ini sudah dipisahkan antara pengambilan data dari web , dan terdapat error handling `response.raise_for_status()`, sehingga jika ada error , program akan berhenti dan tidak akan memproses halaman yang error , sehingga menghindari data yang tidak valid. Full kode setelah refactoring terdapat pada lampiran 12

4.3.6 JunalScaraper.py

Pada file sebelum refactoring (lampiran 13) skrip kode dilakukan *refactoring* untuk memecah logika pengambilan data artikel menjadi fungsi-fungsi kecil yang lebih modular dan mudah dipelihara. Sehingga menghindari terjadinya Long function dan juga terdapat Dulicate Code dalam kode delete.

Sebelum refactorisasi

```
import requests
from bs4 import BeautifulSoup
import mysql.connector
import datetime
from threading import Thread # for multithreading
import time # for rate limiting (optional)
import re

start_time = time.time()

def joint_seminar():
    base_url =
"https://journal.walisongo.ac.id/index.php/PJIS/issue/archive"
    cursor.execute("INSERT INTO `jurnal` (`id_jurnal`,
`nama_jurnal`, `link_jurnal`, `jadwal_terbit`,
`archive_jurnal`, `volume_terbaru`, `link_indeks`,
`indeks_jurnal`) VALUES ('30', 'Joint International
Seminar',
'https://journal.walisongo.ac.id/index.php/PJIS',
'Unknown',
'https://journal.walisongo.ac.id/index.php/PJIS/issue/archive', NULL,
'https://sinta.kemdikbud.go.id/journals/index/?q=Joint+International+Seminar', 'Not Found');")
    db.commit()
```

```

        DELETE a
        FROM artikel AS a
        INNER JOIN (
            SELECT link_artikel, MIN(id_artikel) AS
min_id
            FROM artikel
            GROUP BY link_artikel
        ) AS b ON a.link_artikel = b.link_artikel AND
a.id_artikel <> b.min_id
        """

        cursor.execute(delete_query)
        db.commit()

        page = requests.get(base_url)
        soup = BeautifulSoup(page.content, "html.parser")

        # Find all issue links (assuming they are within
        anchor tags)
        issue_element = soup.find("div",
        class_=("issueDescription", "issueDescriptionImage",
        "issueCoverImage"))
        issue_link = issue_element.find("a")["href"]
        latest_volume = "2015"

        # Update volume_terbaru in jurnal table for the
        current archive_jurnal
        update_query = """
            UPDATE jurnal
            SET volume_terbaru = %s
            WHERE archive_jurnal = %s
        """
        cursor.execute(update_query, (latest_volume,
        base_url))
        db.commit()

        print(f"Latest volume for {base_url}:
        {latest_volume}") # Confirmation message

        archive_jurnal_id = int(30)

        try:
            page_article = requests.get(issue_link)
            soup2 = BeautifulSoup(page_article.content,

```

```

"html.parser")

    # Try scraping from TOC image link first
    toc_image_element = soup2.find("div",
id="issueCoverImage")
    if toc_image_element and
toc_image_element.find("a", href=True):
        toc_image_link =
toc_image_element.find("a")["href"]
        toc_page =requests.get(toc_image_link)
        soup3 = BeautifulSoup(toc_page.content,
"html.parser")

        # Get current date
        current_date = datetime.date.today()

        # Format the date as YYYY-MM-DD
        formatted_date = current_date.strftime('%Y-%m-
%d')

        penulis_artikel = "Unknown"
        artikel_parent = soup3.find_all("div",
class_="tocTitle")

        for detail in artikel_parent:
            nama_artikel = (
                detail.find("a", href=True).text
                if detail.find("a", href=True)
                else "Front Matter"
            )
            link_artikel = (
                detail.find("a", href=True)["href"]
                if detail.find("a", href=True)
                else "#"
            )

            insert_query = """
                INSERT INTO artikel (nama_artikel,
asal_artikel, link_artikel, tanggal_terbit,
penulis_artikel)
                VALUES (%s, %s, %s, %s, %s)
            """
            cursor.execute(insert_query, (nama_artikel,
archive_jurnal_id, link_artikel, formatted_date,
penulis_artikel))
            ..

```

```

        db.commit()

        print(f"{nama_artikel} {archive_jurnal_id}
{link_artikel} {formatted_date} {penulis_artikel}")

        num_articles = len(artikel_parent)

        print("")
        print(f"Number of articles in this issue:
{num_articles}")
        print("-" * 50) # Separator between
issues

    except requests.exceptions.RequestException as e:
        print(f"Error fetching issue link: {e}")
        pass

def RJPTD():
    base_url =
"https://journal.walisongo.ac.id/index.php/rjtpd/issue
/archive"
    cursor.execute("INSERT INTO `jurnal` (`id_jurnal`,
`nama_jurnal`, `link_jurnal`, `jadwal_terbit`,
`archive_jurnal`, `volume_terbaru`, `link_indeks`,
`indeks_jurnal`) VALUES ('45', 'Research Journal on
Teacher Professional Development',
'https://journal.walisongo.ac.id/index.php/rjtpd',
'Twice a year',
'https://journal.walisongo.ac.id/index.php/rjtpd/issue
/archive', NULL,
'https://sinta.kemdikbud.go.id/journals/index/?q=Resea
rch+Journal+on+Teacher+Professional+Development', 'Not
Found');")
    db.commit()

    delete_query = """
        DELETE a
        FROM artikel AS a
        INNER JOIN (
            SELECT link_artikel, MIN(id_artikel) AS
min_id
            FROM artikel
            GROUP BY link_artikel
        ) AS b ON a.link_artikel = b.link_artikel AND

```

Dalam kode ini terlalu banyak menangani banyak tugas sehingga susah untuk dipelihara dan dipahami. Terdapat banyak Duplicate Code dalam eksekusi query dan scraping

```
delete_query = """
    DELETE a
    FROM artikel AS a
    INNER JOIN (
        SELECT link_artikel, MIN(id_artikel) AS min_id
        FROM artikel
        GROUP BY link_artikel
    ) AS b ON a.link_artikel = b.link_artikel AND
    a.id_artikel <> b.min_id
    """

cursor.execute(delete_query)
db.commit()
```

Terdapat pengulangan kode delete pada file kode sebelum refactoring (Duplicate Code)

```
import requests
from bs4 import BeautifulSoup
import mysql.connector

def fetch_page_content(url):
    """Fetches and returns the HTML content of a given
    URL."""
    response = requests.get(url)
    response.raise_for_status()
    return BeautifulSoup(response.content,
        "html.parser")
```

Sesudah Refaktorisasi

```
def scrape_journal_accreditation(search_url):
    """Scrapes accreditation data for journals affiliated
    with UIN Walisongo from SINTA."""
    soup = fetch_page_content(search_url)
    journals = []

    for listing in soup.find_all("div", class_="list-item
row mt-3"):
        affiliation = listing.find("div", class_=lambda
x: x and "affil-loc" in x)
        if affiliation and "Walisongo" in
affiliation.text:
            journal_name = listing.find("div",
class_="affil-name mb-3")
            journal_name =
journal_name.find("a").text.strip() if journal_name else
None

            accreditation = listing.find(
                "span", class_=lambda x: x and
"accredited" in x
            )
            accreditation = (
                accreditation.text.strip().split()[0] if
accreditation else None
            )

            journals.append({"name": journal_name,
"accreditation": accreditation})

    return journals

def scrape_sinta_index(search_url):
    """Scrapes the Sinta index value from a journal
    profile page."""
    soup = fetch_page_content(search_url)
    sinta_card = soup.find("div", class_="card bg-card-
orange")

    if sinta_card:
        sinta_value = sinta_card.find("div",
```

```

def scrape_sinta_index(search_url):
    """Scrapes the Sinta index value from a journal
    profile page."""
    soup = fetch_page_content(search_url)
    sinta_card = soup.find("div", class_="card bg-card-
orange")

    if sinta_card:
        sinta_value = sinta_card.find("div",
class_="stat-num").text.split()[1]
        return f"S{sinta_value}"
    return None

def update_journal_accreditation(cursor, journal_id,
accreditation):
    """Updates the journal accreditation in the
    database."""
    cursor.execute(
        "UPDATE jurnal SET indeks_jurnal = %s WHERE
id_jurnal = %s",
        (accreditation, journal_id),
    )

def process_journals(cursor, journals):
    """Processes journals and updates their
    accreditation in the database."""
    for idx, (journal_id, url) in enumerate(journals,
start=1):
        print(f"Processing journal {idx}: ", end="")

        journals_data =
scrape_journal_accreditation(url)
        if journals_data:
            for journal in journals_data:
                if journal["accreditation"]:
                    update_journal_accreditation(
                        cursor, journal_id,
journal["accreditation"])
            print(
                f"Accreditation:
{journals_data[0]['accreditation']}, ID: {journal_id}"
            )

```

Dengan *refactoring* ini, logika scraping data artikel dipecah menjadi beberapa fungsi terpisah yang membuat code lebih modular. Seperti `connect_to_database()`, `execure_query()`, `scraper_articles()`, dan `delete_duplicaters()`.

```
def delete_duplicates(cursor):
    delete_query = """
        DELETE a FROM artikel AS a
        INNER JOIN (
            SELECT link_artikel, MIN(id_artikel) AS min_id
            FROM artikel
            GROUP BY link_artikel
        ) AS b ON a.link_artikel = b.link_artikel AND
a.id_artikel <> b.min_id
    """
    execute_query(cursor, delete_query)
```

Dibuatnya function Delete agar menghindari Duplicate Code. Full kode setelah refactoring dapat dilihat pada lampiran 14.

4.4 Pengukuran Nilai *Maintainability Index* sesudah diterapkannya *clean code*

Setelah dilakukannya refactoring code sesuai dengan *clean code* pada setiap filenya , didapatkan nilai-nilai matrik LOC, CC dan HV sebagai berikut.

4.4.1 *Line of Code*

Jumlah baris kode dalam setiap file menunjukkan ukuran aplikasi dan dapat mempengaruhi pemeliharaan serta pengembangan di masa depan.

Untuk menghitung nilai LOC, telah dibuat sebuah skrip Python yang secara otomatis menghitung jumlah baris kode dalam setiap file Python dan PHP.

Berikut adalah skrip untuk menghitung LOC pada file Py

```
def calculate_metrics(file_path):
    """Menghitung LOC, Cyclomatic Complexity, dan Halstead
    Volume"""
    code = read_file(file_path)
    if code is None:
        return None

    loc = count_loc_without_comments(code)

    # Menghitung kompleksitas siklomatis (Cyclomatic
    Complexity)
    cc_result = cc_visit(code)
    cc = sum(block.complexity for block in cc_result)

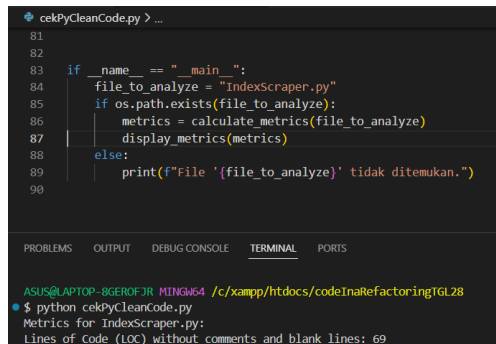
    # Menghitung Halstead Volume
    try:
        hv = h_visit(code).total.volume
    except Exception as e:
        print(f"Error calculating Halstead metrics: {e}")
        hv = 0

    return {"file": os.path.basename(file_path), "loc":
loc, "cc": cc, "hv": hv}

if __name__ == "__main__":
    file_to_analyze = "IndexScraper.py"
    if os.path.exists(file_to_analyze):
        metrics = calculate_metrics(file_to_analyze)
        display_metrics(metrics)
    else:
        print(f"File '{file_to_analyze}' tidak
ditemukan.")
```

Pada Skrip diatas dirancang agar tidak menghitung baris yang berisi komentar (diawali dan diakhiri dengan tanda #,""" ,atau ""') serta baris kosong.

Hasil



The image shows a code editor window with a dark theme. The top part displays Python code for a script named 'cekPyCleanCode.py'. The code includes a main function that checks if a file 'IndexScraper.py' exists, calculates metrics, and displays them. The bottom part of the window shows a terminal output with the command 'python cekPyCleanCode.py' and the result 'Metrics for IndexScraper.py: Lines of Code (LOC) without comments and blank lines: 69'.

```
cekPyCleanCode.py > ...
81
82
83 if __name__ == "__main__":
84     file_to_analyze = "IndexScraper.py"
85     if os.path.exists(file_to_analyze):
86         metrics = calculate_metrics(file_to_analyze)
87         display_metrics(metrics)
88     else:
89         print(f"File '{file_to_analyze}' tidak ditemukan.")
90
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

ASUS@LAPTOP-BGEROFJR MINGW64 /c/xampp/htdocs/codeInaRefactoringTGL28

```
$ python cekPyCleanCode.py
Metrics for IndexScraper.py:
Lines of Code (LOC) without comments and blank lines: 69
```

gambar 4. 8 Hasil Pehitungan LOC File Py Sesudah Refactoring

Berikut adalah skrip untuk menghitung LOC pada file PHP

```
def count_loc_without_comments(code):  
    """Counts lines of code (LOC) excluding comments and  
    blank lines."""  
    lines = code.splitlines()  
    loc = 0  
    in_multiline_comment = False  
  
    for line in lines:  
        stripped_line = line.strip()  
  
        if not stripped_line or  
stripped_line.startswith("//"):  
            continue  
  
        if stripped_line.startswith("/*"):  
            in_multiline_comment = True  
  
        if stripped_line.endswith("*/"):  
            in_multiline_comment = False  
            continue  
  
        if in_multiline_comment:  
            continue  
  
        loc += 1  
  
    return loc  
  
if __name__ == "__main__":  
    file_to_analyze = "Halaman_Artikel.php"  
    if os.path.exists(file_to_analyze):  
        metrics = analyze_php_file(file_to_analyze)  
        if metrics:  
            display_metrics(file_to_analyze, *metrics)  
    else:  
        print(f"File '{file_to_analyze}' not found.")
```

Pada Skrip diatas dirancang agar tidak menghitung baris yang berisi komentar (diawali dengan tanda `//` atau `/`) serta baris kosong.

Hasil

The screenshot shows a code editor with a Python script named `cekPhpCleanCode.py`. The script is designed to analyze a PHP file and calculate the Lines of Code (LOC) without comments and blank lines. The script includes a `__main__` block that sets `file_to_analyze` to `"Halaman_Jurnal.php"`, checks if the file exists, and then calls `analyze_php_file` to calculate the metrics. If metrics are found, they are displayed; otherwise, a message is printed indicating the file was not found.

```

117
118 if __name__ == "__main__":
119     file_to_analyze = "Halaman_Jurnal.php"
120     if os.path.exists(file_to_analyze):
121         metrics = analyze_php_file(file_to_analyze)
122         if metrics:
123             display_metrics(file_to_analyze, *metrics)
124     else:
125         print(f"file '{file_to_analyze}' not found.")
126

```

The terminal output shows the command `python cekPhpCleanCode.py` being executed, resulting in the output: `Metrics for Halaman_Jurnal.php: Lines of Code (LOC) without comments and blank lines: 185`.

gambar 4. 9 Hasil Perhitungan LOC File PHP Sesudah Refactoring

Skrip *full* akan dilampirkan dalam lampiran 1. Berikut adalah hasil LOC sesudah *refactoring* berdasarkan prinsip *clean code* pada setiap file yang diuji dapat dilihat pada tabel :

Tabel 4. 5 LOC Sesudah Refactoring

File	LOC (sesudah)
IndexScrapper.py	69
JurnalScrapper.py	115
Halaman_Artikel.php	274

Index.php	66
Halaman_Home.php	82
Halaman_Jurnal.php	185
Total Rata-Rata	131.833

4.4.2 Cyclomatic Compaxity (CC)

Rumus untuk menghitung Cyclomatic Complexity adalah sebagai berikut:

$$CC=E-N+2P$$

Keterangan :

CC = Cyclomatic complexity

E = Jumlah edge (sisi) dalam graf kontrol alur

N = Jumlah node (simpul),

P = Jumlah komponen terhubung dalam graf

Untuk menghitung nilai CC, telah dibuat sebuah skrip Python yang secara otomatis menghitung jumlah baris kode dalam setiap file Python dan file PHP.

**Berikut adalah Skrip untuk menghitung nilai CC
pada file Python**

```
import os
from radon.complexity import cc_visit
from radon.metrics import h_visit

def calculate_metrics(file_path):
    """Menghitung LOC, Cyclomatic Complexity, dan
    Halstead Volume"""
    code = read_file(file_path)
    if code is None:
        return None

    loc = count_loc_without_comments(code)

    # Menghitung kompleksitas siklomatis (Cyclomatic
    Complexity)
    cc_result = cc_visit(code)
    cc = sum(block.complexity for block in cc_result)

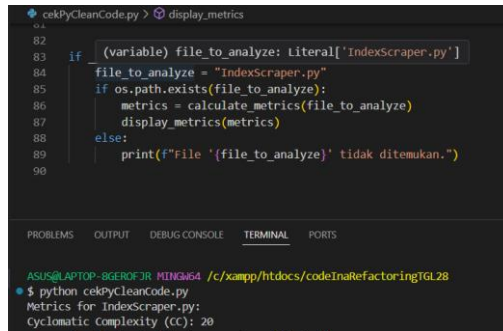
    # Menghitung Halstead Volume
    try:
        hv = h_visit(code).total.volume
    except Exception as e:
        print(f"Error calculating Halstead metrics: {e}")
        hv = 0

    return {"file": os.path.basename(file_path), "loc":
loc, "cc": cc, "hv": hv}

if __name__ == "__main__":
    file_to_analyze = "IndexScraper.py"
    if os.path.exists(file_to_analyze):
        metrics = calculate_metrics(file_to_analyze)
        display_metrics(metrics)
    else:
        print(f"File '{file_to_analyze}' tidak
ditemukan.")
```

Pada skrip ini memakai Library random sehingga kita hanya memanggil module random.complexity dan mengimport cc_visit untuk melakukan perhitungan nilai CC

Hasil



```
cekPyCleanCode.py > display_metrics
82
83 if (variable) file_to_analyze: Literal['IndexScraper.py']
84     file_to_analyze = "IndexScraper.py"
85     if os.path.exists(file_to_analyze):
86         metrics = calculate_metrics(file_to_analyze)
87         display_metrics(metrics)
88     else:
89         print(f"File '{file_to_analyze}' tidak ditemukan.")
90
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ASUSLAPTOP-BEERDFJR MINGW64 /c/xampp/htdocs/codeInaRefactoringTGL28
$ python cekpycleancode.py
Metrics for IndexScraper.py:
Cyclomatic Complexity (CC): 20
```

gambar 4. 10 Hasil Perhitungan CC File Py Sesudah Refactoring

**Berikut adalah skrip untuk menghitung CC
Pada file PHP.**

```
# Constants
INITIAL_CYCLOMATIC_COMPLEXITY = 1
PHP_OPERATORS_PATTERN = (
    r"[=+\-
    */<>!&|?:%]+|\b(if|else|for|foreach|while|switch|case|
    catch|return)\b"
)
PHP_OPERANDS_PATTERN = r"\b[a-zA-Z_][a-zA-Z0-9_]*\b"
CONTROL_FLOW_KEYWORDS = [
    "if",
    "else if",
    "for",
    "foreach",
    "while",
    "switch",
    "case",
    "catch",
]

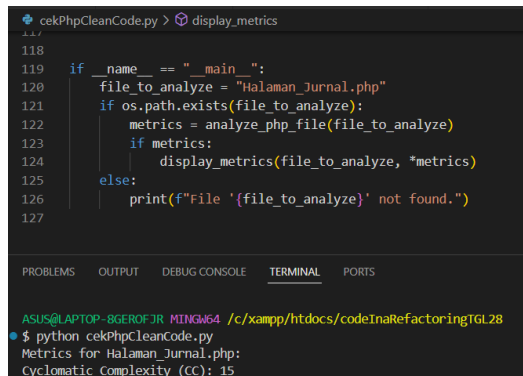
def calculate_cyclomatic_complexity(code):
    """Calculates Cyclomatic Complexity (CC) based on
    control flow keywords."""
    cc = INITIAL_CYCLOMATIC_COMPLEXITY
    for keyword in CONTROL_FLOW_KEYWORDS:
        cc += len(re.findall(rf"\b{keyword}\b", code))
    return cc

if __name__ == "__main__":
    file_to_analyze = "Halaman_Artikel.php"
    if os.path.exists(file_to_analyze):
        metrics = analyze_php_file(file_to_analyze)
```

Pada skrip ini function `calculate_cyclomatic_complexity` berfungsi

mencari kata kunci kontrol alur dalam kode php dan menjumlah kata kunci kontrol alur seperti `if`, `for`, `while`, `switch`, and `case` untuk menentukan nilai CC.

Hasil



The screenshot shows a code editor with a Python script named `cekPhpCleanCode.py` and its terminal output. The script defines a function `display_metrics` and a main block that checks if a file named `Halaman_Jurnal.php` exists. If it exists, it calls `analyze_php_file` and then `display_metrics`. The terminal output shows the command `python cekPhpCleanCode.py` being executed, and the output is `Metrics for Halaman_Jurnal.php: Cyclomatic Complexity (CC): 15`.

```
118
119 if __name__ == "__main__":
120     file_to_analyze = "Halaman_Jurnal.php"
121     if os.path.exists(file_to_analyze):
122         metrics = analyze_php_file(file_to_analyze)
123         if metrics:
124             display_metrics(file_to_analyze, *metrics)
125     else:
126         print(f"File '{file_to_analyze}' not found.")
127
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
ASUS@LAPTOP-8GEROF3R MINGW64 /c:/xampp/htdocs/codeInaRefactoringTGL28
$ python cekPhpCleanCode.py
Metrics for Halaman_Jurnal.php:
Cyclomatic Complexity (CC): 15
```

gambar 4. 11 Hasil Perhitungan CC File PHP Sesudah Refactoring

Skrip *full* akan dilampirkan dalam lampiran 1.

Berikut adalah nilai CC sesudah dilakukannya refactoring sesuai dengan prinsip clean code pada masing-masing file.

Tabel 4. 6 CC Sesudah Refactoring

File	CC (sesudah)
IndeScraper.py	20
JurnalScraper.py	24
Halaman_Artikel.php	31
Index.php	3
Halaman_Home.php	8
Halaman_Jurnal.php	15
Total Rata-Rata	16.83

4.4.3 Halstead Volume (HV)

Rumus menghitung Halstead Volume sebagai berikut :

$$HV = N \times \log_2(\eta)$$

$N = N_1 + N_2$ (total operator dan operand)

$\eta = \eta_1 + \eta_2$ (jumlah unik operator dan operand)

Untuk menghitung nilai HV, telah dibuat sebuah skrip Python yang secara otomatis menghitung jumlah baris kode dalam setiap file Python dan File php.

```

import os
from radon.complexity import cc_visit
from radon.metrics import h_visit

def calculate_metrics(file_path):
    """Menghitung LOC, Cyclomatic Complexity, dan
    Halstead Volume"""
    code = read_file(file_path)
    if code is None:
        return None

    loc = count_loc_without_comments(code)

    # Menghitung kompleksitas siklomatis (Cyclomatic
    Complexity)
    cc_result = cc_visit(code)
    cc = sum(block.complexity for block in cc_result)

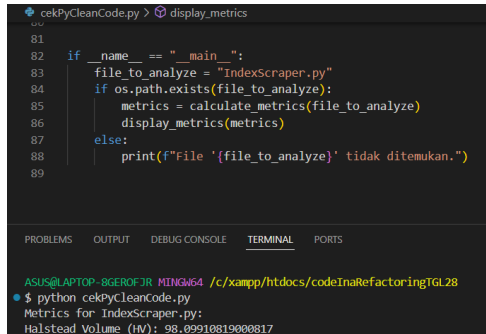
    # Menghitung Halstead Volume
    try:
        hv = h_visit(code).total.volume
    except Exception as e:
        print(f"Error calculating Halstead metrics:
        {e}")
        hv = 0

    return {"file": os.path.basename(file_path), "loc":
    loc, "cc": cc, "hv": hv}

```

Pada skrip ini memakai Library radon sehingga kita hanya memanggil module radon.metrics dan mengimport h_visit untuk melakukan perhitungan nilai HV.

Hasil



```
cekPyCleanCode.py > display_metrics
81
82 if __name__ == "__main__":
83     file_to_analyze = "IndexScraper.py"
84     if os.path.exists(file_to_analyze):
85         metrics = calculate_metrics(file_to_analyze)
86         display_metrics(metrics)
87     else:
88         print(f"File '{file_to_analyze}' tidak ditemukan.")
89
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
ASUS@LAPTOP-8GEROFJR MINGW64 /c/xampp/htdocs/codeInaRefactoringTGL28
$ python cekPyCleanCode.py
Metrics for IndexScraper.py:
Halstead Volume (HV): 98.09910819000817
```

gambar 4. 12 Hasil HV File Py Sesudah Refactoring

Berikut adalah Skrip untuk menghitung nilai HV pada file PHP.

```
# Constants
INITIAL_CYCLOMATIC_COMPLEXITY = 1
PHP_OPERATORS_PATTERN = (
    r"[+=\-\
*</>!&|?:%]+|\b(if|else|for|foreach|while|switch|case|ca
tch|return)\b"
)
PHP_OPERANDS_PATTERN = r"\b[a-zA-Z_][a-zA-Z0-9_]*\b"
CONTROL_FLOW_KEYWORDS = [
    "if",
    "else if",
    "for",
    "foreach",
    "while",
    "switch",
    "case",
    "catch",
]

def extract_operators_operands(code):
    """Extracts operators and operands from PHP code."""
    operators = re.findall(PHP_OPERATORS_PATTERN, code)
    operands = re.findall(PHP_OPERANDS_PATTERN, code)
    return operators, operands
```

```

def calculate_halstead_volume(code):
    """Calculates Halstead Volume (HV) based on operators and
    operands."""
    operators, operands = extract_operators_operands(code)

    unique_operators = set(operators)
    unique_operands = set(operands)

    total_operators = len(operators)
    total_operands = len(operands)

    n1 = len(unique_operators)
    n2 = len(unique_operands)
    N1 = total_operators
    N2 = total_operands

    n = n1 + n2
    N = N1 + N2

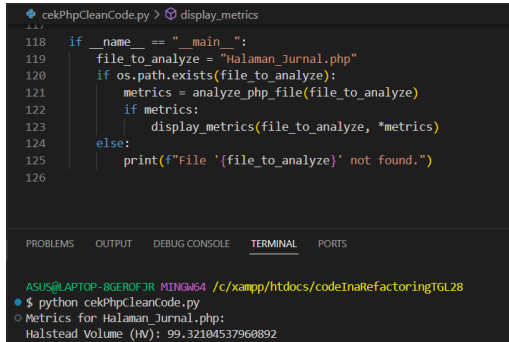
    adjustment_factor = 0.01
    return N * (math.log2(n) * adjustment_factor if n > 1 else
0) if n > 0 else 0

if __name__ == "__main__":
    file_to_analyze = "Halaman_Artikel.php"
    if os.path.exists(file_to_analyze):
        metrics = analyze_php_file(file_to_analyze)
        if metrics:
            display_metrics(file_to_analyze, *metrics)
    else:
        print(f"File '{file_to_analyze}' not found.")

```

Pada skrip ini function `calculate_halstead_volume` merupakan function main untuk menghitung HV, pertama- tama memanggil `extract_operators_operands` untuk menghitung jumlah operator unik (n1), operand unik (n2), total operator (N1), dan total operant (N2) kemudian nilai HV dihitung dengan rumus HV.

Hasil



The screenshot shows a code editor with a Python script named `cekPhpCleanCode.py`. The script is designed to analyze a PHP file and display metrics. The code is as follows:

```
118 if __name__ == "__main__":
119     file_to_analyze = "Halaman_Jurnal.php"
120     if os.path.exists(file_to_analyze):
121         metrics = analyze_php_file(file_to_analyze)
122         if metrics:
123             display_metrics(file_to_analyze, "metrics")
124         else:
125             print(f"File '{file_to_analyze}' not found.")
126
```

Below the code editor, the terminal output is displayed, showing the execution of the script and the resulting metrics for the file `Halaman_Jurnal.php`:

```
ASUS@LAPTOP-BGEROFJR MINGW64 /c:/xampp/htdocs/codeInaRefactoringTGL28
$ python cekPhpCleanCode.py
Metrics for Halaman_Jurnal.php:
Halstead Volume (HV): 99.32104537960892
```

gambar 4. 5 Hasil HV file PHP Sesudah Refactoring

Skrip *full* akan dilampirkan dalam lampiran 1.

Berikut adalah nilai HV setelah dilakukannya refactoring sesuai dengan prinsip clean code pada masing-masing file.

Tabel 4. 7 HV Setelah Refactoring

File	HV (sesudah)
IndexScraper.py	98.09
JurnalScraper.py	24.00
Halaman_Artikel.php	156.38
Index.php	27.91
Halaman_Home.php	45.50
Halaman_Jurnal.php	99.32
Total Rata-Rata	75.2

4.4.4 Nilai *Maintainability Index* (MI) sesudah diterapkan *clean code*.

Adapun Rumus maintainability Index Sebagai berikut :

$$MI = 171 - 5.2 \times \ln(LOC) - 0.23 \times CC - 16.2 \times \ln(HV)$$

Keterangan :

MI: Maintainability Index (Indeks Kemampuan Pemeliharaan)

LOC: Lines of Code (Jumlah Baris Kode)

CC: Cyclomatic Complexity (Kompleksitas Siklomatik)

HV: Volume Halstead

ln: Logaritma natural

Untuk menghitung nilai MI, telah dibuat sebuah skrip Python yang secara otomatis menghitung jumlah baris kode dalam setiap file Python dan File php.

Berikut adalah Skrip untuk menghitung nilai MI File IndexScrapper.py

```
1 import math
2
3
4 def calculate_mi(cc, hv, loc):
5     """
6     Menghitung Maintainability Index (MI) berdasarkan formula standar.
7     Args:
8         cc (int): Cyclomatic Complexity
9         hv (float): Halstead Volume
10        loc (int): Lines of Code
11    Returns:
12        float: Nilai Maintainability Index (MI)
13    """
14    try:
15        mi = 171 - 5.2 * math.log(loc) - 0.23 * cc - 16.2 * math.log(hv)
16        return round(mi, 2)
17    except ValueError as e:
18        return f"Error: {e}. Pastikan nilai LOC dan HV lebih besar dari nol."
19
20
21 # Contoh penggunaan
22 cc = 20 # Cyclomatic Complexity
23 hv = 98.09 # Halstead Volume
24 loc = 69 # Lines of Code
25
26 mi = calculate_mi(cc, hv, loc)
27 print(f"Maintainability Index (MI): {mi}")
28
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
ASUS@LAPTOP-BGEROF3R MINGW64 /c:/xampp/htdocs/codeInaRefactoringTGL28
$ python cekMiCleanCode.py
Maintainability Index (MI): 70.09
```

gambar 4. 6 Perhitungan MI Sesudah Refactoring

Pada skrip ini, perhitungan MI dilakukan dengan memanggil funtion `calculate_mi`, yang menggunakan nilai CC, HV, dan LOC sebagai input. Kemudian skrip ini akan menghitung nilai yang sudah di input dengan rumus MI.

Skrip *full* akan dilampirkan dalam lampiran 2.

Berikut adalah nilai MI setelah dilakukannya refactoring sesuai dengan prinsip clean code pada masing-masing file.

Tabel 4. 8 MI Sesudah Refactoring

File	MI (sesudah)
IndexScraper.py	70.09
JurnalScraper.py	89.19
Halaman_Artikel.php	52.83
Index.php	94.59
Halaman_Home.php	84.40
Halaman_Jurnal.php	67.52
Total rata-rata	76,43

4.5 Analisis Temuan dan Hasil Penelitian.

4.5.1 Perbandingan Nilai MI Sebelum dan Sesudah diterapkannya *Clean code*.

Tabel 4. 9 Perbandingan MI

file	MI (Sebelum Refactoring)	MI (sesudah)
IndexScraper.py	64.02	70.09
JurnalScraper.py	63.7	89.19
Halaman_Artikel.php	28.75	52.83
Index.php	60.79	94.59

Halaman_Home.php	60.75	84.40
Halaman_Jurnal.php	49.42	67.52

Berdasarkan hasil pengukuran data dengan membandingkan Tingkat *maintainability* sebelum dan sesudah *refactoring* didapatkan hasil bahwa terjadinya peningkatan nilai *Maintainability Index* setelah menerapkan prinsip *clean code*

4.5.2 Analisis menggunakan metode komperatif mean standar

Didapatkan hasil sebagai berikut:

Tabel 4. 10 Hasil Uji Mean

No	Metrik	Rata-rata mean Sebelum refactoring	Rata-rata mean sesudah refactoring	Persentase
1	LOC	225.33	131.83	41,49% (turun)
2	CC	25.25	16.83	34,64% (turun)
3	HV	185.71	75,20	59,49%

				(turun)
4	MI	54.57	76,43	40,06% (naik)

Hasil analisis diatas menunjukan bahwa penerapan prinsip clean code memberikan perubahan pada maintaibility kode. Terbukti dengan adanya penurunan tiga metrik utama, yaitu LOC, CC, dan HV. Nilai LOC berkurang dari 225.33 menjadi 131.83, terjadi penurunan sebesar 41.49% yang menunjukan kode menjadi lebih ringkas dan efiesien. CC mengalami penurunan dari 25.25 menjadi 16.67, terjadi penurunan 34.64% yang menandakan bahwa kompleksitas logika dalam kode berkurang, sehingga lebih mudah dipahami dan diuji. Selain itu nilai HV berukurang dari 185.71 menjadi 75.20, terjadi penurunan sebesar 59.49% yang menunjukan bahwa jumlah operasi dan operand dalam kode berkurang, sehingga kompleksitas perhitungan dalam code lebih sederhana.

Dikarenakan penurunan dari ketiga metrik tersebut, nilai Maintaibility index (MI) mengalami peningkatan dari 54.57 menjadi 76.43, atau naik

sebesar 40.06%. Peningkatan ini menunjukkan bahwa kode setelah refactoring lebih mudah dipelihara, lebih terstruktur, dan lebih modular. Secara keseluruhan, hasil ini menunjukkan bahwa penerapan prinsip clean code membantu meningkatkan maintainability code dengan menyederhanakan struktur, memperbaiki keterbacaan, serta menjadikannya lebih efisien untuk pengembangan dan pemeliharaan di masa mendatang.

4.5.3 Analisis Perbandingan dengan Uji -t

Untuk mengetahui apakah terdapat perbedaan signifikan sebelum dan sesudah penerapan prinsip Clean Code, dilakukan pengujian menggunakan uji t berpasangan (Paired Sample t-Test).

1. Hipotesis Penelitian

- a. H_0 : tidak terdapat perbedaan signifikan pada nilai maintainability indeks sebelum dan sesudah penerapan *clean code*
- b. H_1 : terdapat perbedaan signifikan pada nilai maintainability indeks sebelum dan sesudah penerapan *clean code*.

2. Hasil Uji-t

t-Test: Paired Two Sample for Means		
	MI (Sebelum Refactorin	MI (sesudah)
Mean	54.57166667	76.43666667
Variance	188.5470167	246.5335067
Observations	6	6
Pearson Correlation	0.810794007	
Hypothesized Mean Difference	0	
df	5	
t Stat (t-hitung)	-5.793301161	
P(T<=t) one-tail	0.001079239	
t Critical one-tail	2.015048373	
P(T<=t) two-tail	0.002158479	
t Critical two-tail (t-tabel)	2.570581836	

gambar 4. 7 Hasil Uji-t

Berdasarkan hasil perhitungan didapatkan t Stat (t hitung) $(-5.7933) < t$ Critical two-tail

(t-tabel) (2.5705) sehingga H_0 ditolak dan H_1 diterima, sehingga terdapat perbedaan yang signifikan antara nilai maintainability indeks sebelum dan sesudah penerapan prinsip *clean code*.

4.5.4 Pembahasan

1. Analisis Peningkatan *Maintability*

Berdasarkan hasil yang didapatkan , terlihat bahwa penerapan prinsip *clean code* memberikan dampak positif terhadap maintainability kode. Nilai rata-rata *Line of code* (LOC) menurun sebesar 42.10%, Nilai rata

rata *Cyclomatic Complexity* (CC) menurun sebesar 34.64%, nilai rata-rata *Healstead Volume* (HV) penurunan sebesar 59.49%, sehingga nilai *Maintainability Index* (MI) meningkat yang awalnya rendah menjadi sedang dan yang sedang menjadi tinggi setelah *refactoring* dilakukan dengan kenaikan rata rata sebesar 40.06%. Peningkatan ini menunjukkan bahwa kode menjadi lebih mudah dipelihara, yang akan memudahkan pengembang dalam melakukan perubahan dan menambahkan fitur di masa depan.

Selain itu, hasil uji-t dengan metode Paired Two Sample for Means menunjukkan bahwa nilai t hitung $(-5.7933) < t\text{-tabel } (2.5705)$. Sehingga H_0 ditolak dan H_1 diterima, ini menunjukkan bahwa terdapat perbedaan yang signifikan sebelum dan sesudah penerapan prinsip clean code.

Prinsip *clean code* membantu dalam :

- a. Meningkatkan keterbacaan kode , sehingga pengembang lain dapat lebih mudah memahami dan bekerja dengan kode

tersebut. Hal Ditinjau pada gambar yang terdapat pada tahapan 4.3 Melakukan Refactoring (penerapan prinsip clean code)

- b. Mengurangi kompleksitas kode dengan memecah metode panjang dan mengorganisir kelas besar sehingga lebih modular. Langkah ini berkontribusi dalam menurunkan *Cyclomatic Complexity*, *Line of Code*, *Healstead Volume*, dan meningkatkan *Maintainability Index*. Hal Ditinjau pada gambar yang terdapat pada tahapan 4.3 Melakukan Refactoring (penerapan prinsip clean code)
 - c. Menghilangkan *code smell* seperti duplikasi kode dan metode panjang, yang membuat kode lebih bersih dan mudah dikelola.
2. Dampak terhadap pemeliharaan kode di masa depan.

Peningkatan nilai MI dan penurunan CC, LOC , dan HV mencerminkan bahwa kode yang telah direfactoring lebih mudah dipelihara . Kode yang lebih mudah dipelihara berarti :

- a. Waktu dan biaya pemeliharaan yang lebih rendah di masa depan karena pengembang

dapat lebih cepat memahami dan mengubah kode.

- b. Kode yang lebih modular dan terstruktur dengan baik mengurangi resiko *bug*, karena dapat memudahkan pengembang dalam melakukan pengujian dan debugging.

Kemampuan untuk berkembang lebih baik karena kode lebih modular dan dapat dengan mudah diperbarui tanpa mempengaruhi bagian lain dari sistem. Hal Ditinjau pada gambar yang terdapat pada tahapan 4.3 Melakukan Refactoring (penerapan prinsip clean code)

Dengan demikian , penerapan clean code tidak hanya meningkatkan MI tetapi juga secara langsung mempengaruhi kualitas dan efisiensi pemeliharaan kode dimasa depan.

BAB V

KESIMPULAN DAN SARAN

5.1 Saran

1. Bagi pengembang perangkat lunak

Disarankan untuk menerapkan prinsip clean code sejak awal pengembangan agar maintainability tetap tinggi dan tidak membutuhkan refactoring yang kompleks dikemudian hari

2. Bagi peneliti selanjutnya

Disarankan untuk mengkombinasikan prinsip clean code dengan metode lain, seperti Design Patterns atau Software Architecture Principles untuk mendapatkan hasil yang lebih optimal

5.2 Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan mengenai optimasi maintainability aplikasi dengan prinsip clean code pada studi kasus Website Monitoring Sistem Jurnal, dapat disimpulkan beberapa hal sebagai berikut :

- 1. Penerapan prinsip clean code secara signifikan meningkatkan maintainability kode program hal ini dibuktikan dengan hasil nilai t hitung (-5.7933)**

$< t\text{-tabel} (2.5705)$ oleh karena itu ditolaknya H_0 dan H_1 diterima, serta terdapat peningkatan nilai maintainability indeks pada setiap filenya setelah dilakukan refactoring , dengan menaikkan nilai rata-rata Maintability Indeks sebesar 40.13%. Hal ini menunjukkan perbaikan dalam keterbacaan dan kemudahan pemeliharaan code

2. Terdapat penurunan nilai rata-rata Cyclomatic Complexity (CC), Line of Code (CC), dan Halstead Volume (HV) setelah penerapan clean code. Penurunan LOC sebesar 42.10%, CC sebesar 34.64%, dan HV sebesar 59.49%, menunjukkan bahwa kode yang telah di optimasi menjadi lebih ringkas dan efisien sehingga lebih mudah dikelola dan dikembangkan. Prinsip clean code membantu dalam mengurangi code smell, seperti duplikasi kode dan metode yang terlalu panjang. Hal ini membuat struktur kode lebih modular, meningkatkan keterbacaan dan mempermudah proses debugging serta pengembangan fitur baru dimasa depan.

DAFTAR PUSTAKA

- Ardito, L., Coppola, R., Barbato, L., & Verga, D. (2020). A Tool-Based Perspective On Software Code Maintainability Metrics: A Systematic Literature Review. In *Scientific Programming* (Vol. 2020). Hindawi Limited. <https://doi.org/10.1155/2020/8840389>
- Arhandi, P. P., Pramitarini, Y., & Alviandra, R. (2019). Desain Prototype Frontend Auto Generator Based On REST API. In *Seminar Informatika Aplikatif Polinema* (pp. 389-393).
- Alsolai, H., & Roper, M. (2020). A systematic literature review of machine learning techniques for software maintainability prediction. *Information and Software Technology*, 119, 106214.
- Aswini, S., & Yazhini, M. (2017). An assessment framework of routing complexities using LOC metrics. 2017 Innovations in Power and Advanced Computing Technologies, i-PACT 2017, 2017-Janua,1-6. <https://doi.org/10.1109/IPACT.2017.8245022>
- Arif, A., & Rana, Z. A. (2020, December). Refactoring of code to remove technical debt and reduce maintenance effort. In *2020 14th International Conference on Open Source Systems and Technologies (ICOSST)* (pp. 1-7). IEEE.
- Atmaja, R. G., Priyambadha, B., & Pradana, F. (2019). Pembangunan Kakas Bantu Untuk Mengukur Maintainability Index Pada Perangkat Lunak Berdasarkan Nilai Halstead Metrics dan McCabe's Cyclomatic Complexity: English. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(3), 2167-2172.
- Agianto, H. N., & Astuti, T. D. (2023). IMPLEMENTASI SISTEM KEUANGAN KASIR BERBASIS APLIKASI ACCURATE POS PADA UMKM (Studi Kasus pada Usaha Minimarket dan Toko Penjual

- Ikan Laut Jogja). *Community Development Journal: Jurnal Pengabdian Masyarakat*, 4(6), 12445-12452.
- Boutaib, S., Bechikh, S., Palomba, F., Elarbi, M., Makhoul, M., & Said, L. B. (2021). Code smell detection and identification in imbalanced environments. *Expert Systems with Applications*, 166, 114076.
- De Silva, D., Menaka, M., Jethaka, M., Abeyaratne, S. R., & Methmini, M. (2023). Software complexity metrics for real-time systems. *International Journal of Innovative Science and Research Technology*, 8(6), 2026-2030
- di Biase, M., Rastogi, A., Bruntink, M., & van Deursen, A. (2019, May). The delta maintainability model: Measuring maintainability of fine-grained code changes. In *2019 IEEE/ACM International Conference on Technical Debt (TechDebt)* (pp. 113-122). IEEE.
- Dikilitaş, Y., Çakal, Ç., Okumuş, A. C., Yalçın, H. N., Yıldırım, E., Ulusoy, Ö. F., ... Macit, B. (2023). Performance Analysis For Web Scraping Tools: Case Studies On BeautifulSoup, Scrapy, Htmlunit And Jsoup . *International Conference On Emerging Trends And Applications In Artificial Intelligence, Icetai 2023* (Pp.471-480). İstanbul, Turkey
- Fenton, N. E., & Bieman, J. M. (2014). *Software metrics: A rigorous and practical approach* (3rd ed.). CRC Press
- Fowler, M. (2018). *Refactoring: improving the design of existing code*. Addison-Wesley Professional.
- Haddit Azhizi, M., & Yaqin, M. A. (2024). Analisis Penggunaan Pemrograman Berorientasi Obyek Terhadap Maintainability Perangkat Lunak Menggunakan Odoo. *Jacis: Journal Automation Computer Information System*, 4, 50–59. <https://doi.org/10.47134/jacis.v4i2.74>
- Heričko, T., & Šumak, B. (2023). Exploring Maintainability Index Variants For Software Maintainability Measurement In Object-

- Oriented Systems. *Applied Sciences (Switzerland)*, 13(5). <https://doi.org/10.3390/app13052972>
- I.Khaleel, S., & Khaleel Al-Khatouni, G. (2023). Measuring Maintainability Index Before And After Code Refactoring. *Minar International Journal Of Applied Sciences And Technology*, 05(03), 18–36. <https://doi.org/10.47832/2717-8234.16.2>
- Inayah, F. I., & Idris, M. (2021). Implementasi Clean Code Pada Pengembangan Aplikasi Berbasis Web. *AUTOMATA*, 2(2).
- Kurniawan, W., Widhiarso, W., & Rachmadi, M. (2024, May). Implementasi Clean Code Berdasarkan Smells and Heuristics Aplikasi MDP SIGN. In *MDP Student Conference* (Vol. 3, No. 1, pp. 956-965).
- Laksono, W. P., Satria, B., Wicaksana, T., Wijasena, A. Y., & Sahria, Y. (2024b). Implementasi Clean Architecture Dalam Membangun Aplikasi Mobile Menggunakan Flutter. *Nusantara Journal Of Multidisciplinary Science*, 2(1), 173–180. <https://jurnal.intekom.id/index.php/Njms>
- Lenarduzzi, V., Kilamo, T., & Janes, A. (2023). Does Cyclomatic or Cognitive Complexity Better Represents Code Understandability? An Empirical Investigation on the Developers Perception. *ArXiv Preprint ArXiv:2303.07722*.
- Lestari, W. F., Chrisnanto, Y. H., & Yuniarti, R. (2024). Teknik Refactoring untuk Kualitas Usability Sistem Informasi SPP dan Tunggakan SDIT Nuralima. *Reslaj: Religion Education Social Laa Roiba Journal*, 6(8), 4066-4078.
- Manuaba, I. B. K., Abdillah, R., Datya, A. I., Yuliyanti, S., Widiyasono, N., & Bahana, R. (2023). Pengembangan Aplikasi Perangkat Lunak: Panduan Praktis dalam Merancang, Membangun, dan Mengelola Aplikasi. PT. Sonpedia Publishing Indonesia.
- Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall

- Mulyawan, M. D., Kumara, I. N. S., Swamardika, I. B. A., & Saputra, K. O. (2021). **Pengukuran Kualitas Sistem Informasi Berdasarkan ISO/IEC 25010: Literature Review**. *Majalah Ilmiah Teknologi Elektro*, 20(1), 15-27. DOI: <https://doi.org/10.24843/MITE.2021.v20i01.P02>.
- Putra, A. D., Alexandra, W., & Puspaningrum, A. S. (2022). A Penerapan Teknologi Augmented Reality Berbasis Android Untuk Pembelajaran Rantai Makanan Pada Hewan. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 3(1), 1-24.
- Rachmad, Y. E., Tampubolon, L. P. D., Purbaratri, W., Sudipa, I. G. I., Ariana, A. A. G. B., Faried, M. I., ... & Kurniawan, H. (2023). *Rekayasa Perangkat Lunak*. PT. Sonpedia Publishing Indonesia.
- Tampubolon, D. J. A., Ardiansyah, F., Makarim, R. H., Irviantina, S., & Kurniawan, H. (2023). Pengembangan Sistem Informasi Berbasis Web Pada Sekolah SMAN 12 Medan Dengan Metode Extreme Programming. *Jurnal Sifo Mikroskil (JSM)*, 24(2).
- Topuz, F. N. (2022). *Empirical Evidence of the Consequences of Bad Smells in Software* (Doctoral dissertation, Auburn University).
- V. Lenarduzzi, F. Lomio, H. Huttunen and D. Taibi, "Are SonarQube Rules Inducing Bugs?," 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER), London, ON, Canada, 2020, pp. 501-511, doi: 10.1109/SANER48275.2020.9054821.
- Widhiarso, W., & Rachmadi, M. (2024, April). Penerapan Clean code pada Aplikasi Siizin Berdasarkan Smells and Heuristics. In *MDP Student Conference* (Vol. 3, No. 1, pp. 95-102).
- Wijaya, R. A., & Karmilasari, K. (2021). Pengukuran Kualitas Website Pengurus Cabang Nu Depok Menggunakan Software Metric. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 10(3), 438-443. <https://doi.org/10.32736/Sisfokom.v10i3.1267>

Yusup, B. P. (2019). *PEMROGRAMAN PROGRAMMABLE LOGIC CONTROLLER (PLC) BERBASIS "CLEAN CODE"* (Doctoral dissertation, Universitas Widya Dharma).

Lampiran 1 Skrip Menghitung nilai LOC,CC,dan HV untuk file Py

```
import os
from radon.complexity import cc_visit
from radon.metrics import h_visit

def read_file(file_path):
    """Membaca file dan mengembalikan isinya"""
    try:
        with open(file_path, "r", encoding="utf-8") as file:
            return file.read()
    except Exception as e:
        print(f"Error reading file: {e}")
        return None

def count_loc_without_comments(code):
    """Menghitung jumlah baris kode tanpa komentar dan baris kosong"""
    lines = code.splitlines()
    loc = 0
    in_multiline_comment = False

    for line in lines:
        stripped_line = line.strip()

        # Lewati baris kosong
        if not stripped_line:
            continue

        # Deteksi awal dan akhir komentar multiline
        if stripped_line.startswith('"""') or
        stripped_line.startswith('\'\'\''):
            if stripped_line.count('\'\'\'') == 2 or
            stripped_line.count('"""') == 2:
                # Jika komentar dibuka dan ditutup dalam satu baris,
                abaikan baris ini
                continue
            in_multiline_comment = not in_multiline_comment
            continue

        # Jika sedang dalam komentar multiline, lewati
        if in_multiline_comment:
            continue

        # Lewati komentar satu baris
        if stripped_line.startswith("#"):
            continue

        # Hitung sebagai baris kode
        loc += 1

    return loc
```

```

def calculate_metrics(file_path):
    """Menghitung LOC, Cyclomatic Complexity, dan Halstead
    Volume"""
    code = read_file(file_path)
    if code is None:
        return None

    loc = count_loc_without_comments(code)

    # Menghitung kompleksitas siklomatis (Cyclomatic
    Complexity)
    cc_result = cc_visit(code)
    cc = sum(block.complexity for block in cc_result)

    # Menghitung Halstead Volume
    try:
        hv = h_visit(code).total.volume
    except Exception as e:
        print(f"Error calculating Halstead metrics: {e}")
        hv = 0

    return {"file": os.path.basename(file_path), "loc": loc,
            "cc": cc, "hv": hv}

def display_metrics(metrics):
    """Menampilkan hasil analisis metrik"""
    if metrics:
        print(f"Metrics for {metrics['file']}:")
        print(f"Halstead Volume (HV): {metrics['hv']}")
        print(f"Cyclomatic Complexity (CC): {metrics['cc']}")
        print(f"Lines of Code (LOC) without comments and blank
lines: {metrics['loc']}")

if __name__ == "__main__":
    file_to_analyze = "IndexScraper.py"
    if os.path.exists(file_to_analyze):
        metrics = calculate_metrics(file_to_analyze)
        display_metrics(metrics)
    else:
        print(f"File '{file_to_analyze}' tidak ditemukan.")

```

Lampiran 2 Skrip Menghitung nilai LOC,CC,dan HV untuk file PHP

```
import os
import re
import math

# Constants
INITIAL_CYCLOMATIC_COMPLEXITY = 1
PHP_OPERATORS_PATTERN = (
    r"[=+\\-
    */<>!&|?:%]+|\\b(if|else|for|foreach|while|switch|case|catch|return)\\b
    "
)
PHP_OPERANDS_PATTERN = r"\\b[a-zA-Z_][a-zA-Z0-9_]*\\b"
CONTROL_FLOW_KEYWORDS = [
    "if",
    "else if",
    "for",
    "foreach",
    "while",
    "switch",
    "case",
    "catch",
]

def read_file(file_path):
    """Reads the content of a file and returns it as a string."""
    try:
        with open(file_path, "r", encoding="utf-8") as file:
            return file.read()
    except Exception as e:
        print(f"Error reading file: {e}")
        return None

def count_loc_without_comments(code):
    """Counts lines of code (LOC) excluding comments and blank
    lines."""
    lines = code.splitlines()
    loc = 0
    in_multiline_comment = False

    for line in lines:
        stripped_line = line.strip()

        if not stripped_line or stripped_line.startswith("//"):
            continue

        if stripped_line.startswith("/*"):
            in_multiline_comment = True

        if stripped_line.endswith("*/"):
            in_multiline_comment = False
            continue
```

```

        if in_multiline_comment:
            continue

        loc += 1

    return loc

def calculate_cyclomatic_complexity(code):
    """Calculates Cyclomatic Complexity (CC) based on control flow
    keywords."""
    cc = INITIAL_CYCLOMATIC_COMPLEXITY
    for keyword in CONTROL_FLOW_KEYWORDS:
        cc += len(re.findall(rf"\b{keyword}\b", code))
    return cc

def extract_operators_operands(code):
    """Extracts operators and operands from PHP code."""
    operators = re.findall(PHP_OPERATORS_PATTERN, code)
    operands = re.findall(PHP_OPERANDS_PATTERN, code)
    return operators, operands

def calculate_halstead_volume(code):
    """Calculates Halstead Volume (HV) based on operators and
    operands."""
    operators, operands = extract_operators_operands(code)

    unique_operators = set(operators)
    unique_operands = set(operands)

    total_operators = len(operators)
    total_operands = len(operands)

    n1 = len(unique_operators)
    n2 = len(unique_operands)
    N1 = total_operators
    N2 = total_operands

    n = n1 + n2
    N = N1 + N2

    adjustment_factor = 0.01
    return N * (math.log2(n) * adjustment_factor if n > 1 else 0) if
n > 0 else 0

def analyze_php_file(file_path):
    """Analyzes a PHP file and returns LOC, Cyclomatic Complexity,
    and Halstead Volume."""
    code = read_file(file_path)
    if code is None:
        return None

    loc = count_loc_without_comments(code)
    cc = calculate_cyclomatic_complexity(code)
    hv = calculate_halstead_volume(code)

    return loc, cc, hv

```

```

def display_metrics(file_path, loc, cc, hv):
    """Displays calculated metrics for a given file."""
    print(f"Metrics for {os.path.basename(file_path)}:")
    print(f"Halstead Volume (HV): {hv}")
    print(f"Cyclomatic Complexity (CC): {cc}")
    print(f"Lines of Code (LOC) without comments and blank lines:
{loc}")

if __name__ == "__main__":
    file_to_analyze = "Halaman_Jurnal.php"
    if os.path.exists(file_to_analyze):
        metrics = analyze_php_file(file_to_analyze)
        if metrics:
            display_metrics(file_to_analyze, *metrics)
    else:
        print(f"File '{file_to_analyze}' not found.")

```

Lampiran 3 Skrip Kode halaman_Artikel.php sebelum penerapan prinsip *Clean Code*

```
<?php
    include "Koneksi.php";
?>

<!doctype html>
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-
ie8 lt-ie7" lang=""> <![endif]-->
<!--[if IE 7]>        <html class="no-js lt-ie9 lt-
ie8" lang=""> <![endif]-->
<!--[if IE 8]>        <html class="no-js lt-ie9"
lang=""> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js" lang="">
<!--<![endif]-->
<head>
    <meta http-equiv="Content-Type"
content="text/html; charset=utf-8" />
    <!-- <meta http-equiv="refresh" content="180"> -->

    <title>Halaman Artikel</title>
    <meta name="description" content="Monitoring
Jurnal">
    <meta name="viewport" content="width=device-width,
initial-scale=1">

    <link rel="apple-touch-icon"
href="images/uinws.png">
    <link rel="shortcut icon" href="images/uinws.png">

    <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/normalize.css@8.0.0
/normalize.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/bootstrap@4.1.3/dis
t/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/font-
awesome@4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/gh/lykmapipo/themify-
icons@0.1.2/css/themify-icons.css">
```

```

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/pixeden-stroke-7-
icon@1.2.3/pe-icon-7-stroke/dist/pe-icon-7-
stroke.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/flag-
icon-css/3.2.0/css/flag-icon.min.css">
    <link rel="stylesheet" href="assets/css/cs-skin-
elastic.css">
    <link rel="stylesheet"
href="assets/css/style.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.
1/jquery.min.js"></script>
    <!-- <script type="text/javascript"
src="https://cdn.jsdelivr.net/html5shiv/3.7.3/html5shi
v.min.js"></script> -->
    <link
href="https://cdn.jsdelivr.net/npm/chartist@0.11.0/dis
t/chartist.min.css" rel="stylesheet">
    <link
href="https://cdn.jsdelivr.net/npm/jqvmap@1.5.1/dist/j
qvmap.min.css" rel="stylesheet">

    <link
href="https://cdn.jsdelivr.net/npm/weathericons@2.1.0/
css/weather-icons.css" rel="stylesheet" />
    <link
href="https://cdn.jsdelivr.net/npm/fullcalendar@3.9.0/
dist/fullcalendar.min.css" rel="stylesheet" />

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/1.5.
3/jspdf.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf-
autotable/3.5.6/jspdf.plugin.autotable.min.js"></scrip
t>
    <script type="text/javascript"
src="assets/js/pdf/main.js"></script>

</head>

<body>
    <!-- Left Panel -->

```

```

<aside id="left-panel" class="left-panel">
  <nav class="navbar navbar-expand-sm navbar-default">
    <div id="main-menu" class="main-menu collapse navbar-
collapse">
      <ul class="nav navbar-nav">
        <li class="">
          <a href="Halaman_Home.php"><i class="menu-icon fa
fa-home"></i>Halaman Home </a>
        </li>
        <li class="">
          <a href="Halaman_Jurnal.php"><i class="menu-icon
fa fa-laptop"></i>Halaman Jurnal</a>
        </li>
        <li class="">
          <a href="Halaman_Artikel.php"><i class="menu-icon
fa fa-book"></i>Halaman Artikel</a>
        </li>
      </ul>
    </div><!-- /.navbar-collapse -->
  </nav>
</aside>
<!-- /#left-panel -->
<!-- Right Panel -->
<div id="right-panel" class="right-panel">
  <!-- Header-->
  <header id="header" class="header">
    <div class="top-left">
      <div class="navbar-header">
        <a class="navbar-brand" href=""></a>
        <a class="navbar-brand hidden" href=""></a>
        <a id="menuToggle" class="menutoggle"><i class="fa
fa-bars"></i></a>
      </div>
    </div>
  </header>
  <!-- /#header -->
  <!-- Content -->
  <div class="content">

```



```

<?php
                                $yearResult
= mysqli_query($koneksi, $yearQuery); // Reuse the same
query

$selectEndYear = isset($_GET['end_year']) ?
$_GET['end_year'] : 'none';
                                while ($row
= mysqli_fetch_assoc($yearResult)) {

$select = ($row['year'] == $selectEndYear) ? 'selected'
: '';

                                echo
'<option value="" . $row['year'] . " " . $select . ">" .
$row['year'] . "</option>";
                                }
                                ?>
                                </select>
                                </div>
                                </div>
                                <div class="form-row">
                                <div class="form-group
col-md-4">

                                <label
for="sortFilter">Sort By:</label>

                                <select
class="form-control" id="sortFilter" name="sort_by">
                                <option
value="none" <?php echo isset($_GET['sort_by']) &&
$_GET['sort_by'] === 'none' ? 'selected' : '';
?>>None</option>

                                <option
value="nama" <?php echo isset($_GET['sort_by']) &&
$_GET['sort_by'] === 'nama' ? 'selected' : '';
?>>Nama</option>

                                <option
value="tanggal_terbit" <?php echo isset($_GET['sort_by'])
&& $_GET['sort_by'] === 'tanggal_terbit' ? 'selected' : '';
?>>Tanggal Terbit</option>

                                </select>
                                </div>
                                <div class="form-group
col-md-4">

                                <label
for="jurnalFilter">Journal Filter:</label>

                                <select
class="form-control" id="jurnalFilter" name="jurnal">
                                <option
value="none">None</option>

                                $jurnalQuery =

```

```

<?php
    $jurnalQuery
= "SELECT id jurnal, nama jurnal FROM jurnal ORDER BY
nama_jurnal ASC";
    $jurnalResult
= mysqli_query($koneksi, $jurnalQuery);

    $selectedJurnal = isset($_GET['jurnal']) ?
    $_GET['jurnal'] : 'none';

    while ($row =
mysqli_fetch_assoc($jurnalResult)) {
        $selected
= ($row['nama_jurnal'] == $selectedJurnal) ? 'selected' :
'';

        echo
'<option value="" . $row['nama_jurnal'] . ' ' .
$selected . '>' . $row['nama_jurnal'] . '</option>';
    }
    ?>
</select>
</div>
<div class="form-
group col-md-4">
    <label
for="keywordSearch">Keyword Search:</label>
    <input
type="text" class="form-control" id="keywordSearch"
name="keyword" placeholder="Enter keywords..."
value="<?php echo isset($_GET['keyword']) ?
$_GET['keyword'] : ''; ?>">
    </div>
</div>
</div>
</div>
</div> <!-- /.col-lg-12 -->
</div>
<!-- /.orders -->
</div>
<!-- Animated -->
<div class="animated fadeIn">
    <div class="clearfix"></div>
    <!-- Orders -->
    <div class="orders">
        <div class="row">
            <div class="col-lg-12">
                <div class="card">
                    <div class="card-header">

```

```

text-center">Artikel Ilmiah</h3>
                                </div>
                                <div class="card-body">
                                    <div class="table-
stats order-table ov-h">
                                <table
class="table" id="table-1">
                                <thead>
                                    <tr>

<th>No</th>

<th>Nama Artikel</th>

<th>Tanggal Terbit</th>

<th>Asal Artikel</th>

                                </tr>
                                </thead>
                                <tbody>
                                    <?php
                                    //
Pagination Variables
                                $limit =
20; // Number of items per page
                                $page =
isset($_GET['page']) && is_numeric($_GET['page']) ?
$_GET['page'] : 1; // Current page, default to 1 if not
valid
                                $start =
($page - 1) * $limit; // Starting index for fetching data
                                //
Default query to fetch all articles
                                $query =
"SELECT * FROM artikel";
                                // Handle
filters
$conditions = [];

                                // Start
Year filter
                                if
(isset($_GET['start_year']) && $_GET['start_year'] !==
'none') {

$selectedStartYear = mysqli_real_escape_string($koneksi,
$_GET['start_year']);

```

```

$conditions[] = "YEAR(tanggal_terbit) >=
$selectedStartYear";
    }

    // End
Year filter
    if
    (isset($_GET['end_year']) && $_GET['end_year'] !==
'none') {

$selectedEndYear = mysqli_real_escape_string($koneksi,
$_GET['end_year']);

$conditions[] = "YEAR(tanggal_terbit) <=
$selectedEndYear";
    } else if
    (isset($_GET['start_year']) && $_GET['start_year'] !==
'none') {
    // If
    only start year is selected, show articles from start
    year to max year

    $maxYearQuery = "SELECT MAX(YEAR(tanggal_terbit)) as
max_year FROM artikel";

    $maxYearResult = mysqli_query($koneksi, $maxYearQuery);

    $maxYear =
    mysqli_fetch_assoc($maxYearResult)['max_year'];

    $conditions[] = "YEAR(tanggal_terbit) <= $maxYear";
    }

    //
Journal filter
    if
    (isset($_GET['jurnal']) && $_GET['jurnal'] !== 'none') {

$selectedJurnal = mysqli_real_escape_string($koneksi,
$_GET['jurnal']);

$jurnalIdQuery = "SELECT id_jurnal FROM jurnal WHERE
nama_jurnal = '$selectedJurnal'";

$jurnalIdResult = mysqli_query($koneksi, $jurnalIdQuery);
    if
    ($jurnalIdRow = mysqli_fetch_assoc($jurnalIdResult)) {

$jurnalId = $jurnalIdRow['id_jurnal'];

```

```

$conditions[] = "asal_artikel = $jurnalId";
    }

    //
    Keyword search
    $keyword

    = '';

    if
    (isset($_GET['keyword']) && $_GET['keyword'] !== '') {

    $keyword = mysqli_real_escape_string($koneksi,
    $_GET['keyword']);

    $conditions[] = "(nama_artikel LIKE '%$keyword%' OR
    penulis_artikel LIKE '%$keyword%')";
    }

    //
    Combine conditions
    if
    (!empty($conditions)) {

    $query .= " WHERE " . implode(" AND ", $conditions);
    }

    // Handle
    sorting
    $sort_by

    = isset($_GET['sort_by']) ? $_GET['sort_by'] : 'nama'; //
    Default sort by 'nama'

    $sort_order = 'ASC'; // Default order

    if
    ($sort_by === 'tanggal_terbit') {

    $sort_order = 'DESC'; // Sort by 'tanggal_terbit'
    descending
    }

    // Add
    order and limit clause
    $query .=

    " ORDER BY ";

    if
    ($sort_by === 'nama') {

    $query .= "nama_artikel ASC";

    } elseif

```

```

($sort_by === 'tanggal_terbit') {

$query .= "tanggal_terbit $sort_order";
}
$query .=

" LIMIT $start, $limit";

// Fetch
data from database
$result =
mysqli_query($koneksi, $query);

// Check
if the query was successful
if
($result) {
//
Display fetched data
$no =
$start + 1;
while
($data = mysqli_fetch_array($result)) {
?>

<tr>

<td><?php echo $no++; ?></td>

<td><a class="nama_artikel" href="<?php echo
$data['link_artikel']; ?>"><?php echo
$data['nama_artikel']; ?></a>

<br>

<br>

Penulis: <?php echo $data['penulis_artikel']; ?>

</td>

<td><span class="tanggal_terbit"><?php echo
$data['tanggal_terbit']; ?></span></td>

<td>

<span class="asal_artikel">

<?php

$idtest = $data['asal_artikel'];

```

```

$str = strval($idtest);

$sasal = mysqli_query($koneksi, "SELECT nama_jurnal FROM
jurnal WHERE id_jurnal = $str");

$sasal_artikel = mysqli_fetch_assoc($sasal);

$sasal_final = implode(" ", $sasal_artikel);

echo $sasal_final;

?>

</span>

</td>

</tr>

<?php
    }
    } else {
        echo
" <tr><td colspan='4'>No data found</td></tr>";
    }
?>
</tbody>
</table>
</div> <!-- /.table-

stats -->

<div
class="pagination justify-content-center">
    <?php
        $total_query =
"SELECT COUNT(*) as total FROM artikel";
        if
(!empty($conditions)) {
            $total_query
            .= " WHERE " . implode(" AND ", $conditions);
        }
        $total_result =
mysqli_query($koneksi, $total_query);

        // Check if the
        query was successful
        if
        ($total_result) {
            $total_data =
mysqli_fetch_assoc($total_result)['total'];
            $total_pages

```

```

= ceil($total_data / $limit);

// Previous
page link
if ($page >
1) {
    echo '<a
class="page-link" href="?page=' . ($page - 1) .
'&start_year=' . $selectedStartYear . '&end_year=' .
$selectedEndYear . '&jurnal=' . $selectedJurnal .
'&keyword=' . $keyword . '">Previous</a>';
}

$start_page =
max(1, $page - 2); // Adjust starting page based on
current page
$end_page =
min($total_pages, $page + 2); // Adjust ending page based
on current page and total pages

// Display
first 3 page links
for ($i =
$start_page; $i < $page; $i++) {
    $active_class = ($page == $i) ? ' active' : '';
    echo '<a
class="page-link" . $active_class . '" href="?page=' . $i
. '&start_year=' . $selectedStartYear . '&end_year=' .
$selectedEndYear . '&jurnal=' . $selectedJurnal .
'&keyword=' . $keyword . '">' . $i . '</a>';
}

// Display
current page link
echo '<a
class="page-link active" href="?page=' . $page .
'&start_year=' . $selectedStartYear . '&end_year=' .
$selectedEndYear . '&jurnal=' . $selectedJurnal .
'&keyword=' . $keyword . '">' . $page . '</a>';

// Display
last 2 page links (if applicable)
if ($end_page
> $page) {
    for ($i =
$page + 1; $i <= $end_page; $i++) {
        $active_class = ($page == $i) ? ' active' : '';
        echo
'<a class="page-link" . $active_class . '" href="?page=' .

```

```

                                echo
'<a class="page-link' . $active_class . '" href="?page='
. $i . '&start_year=' . $selectedStartYear . '&end_year='
. $selectedEndYear . '&jurnal=' . $selectedJurnal .
'&keyword=' . $keyword . '">' . $i . '</a>';
                                }
                                }

                                // Next page
link
                                if ($page <
$total_pages) {
                                echo '<a
class="page-link" href="?page=' . ($page + 1) .
'&start_year=' . $selectedStartYear . '&end_year=' .
$selectedEndYear . '&jurnal=' . $selectedJurnal .
'&keyword=' . $keyword . '">Next</a>';
                                }
                                }
                                ?>
                                </div>

                                <div id="jump-to-
page" style="display: always; margin-top: 5px;">
                                <form action=""
method="get">
                                <label
for="jump_page">Jump to page:</label>
                                <input
type="number" id="jump_page" name="page" min="1"
max="<?php echo $total_pages; ?>" value="1">
                                <input
type="hidden" name="start_year" value="<?php echo
$selectedStartYear; ?>">
                                <input
type="hidden" name="end_year" value="<?php echo
$selectedEndYear; ?>">
                                <input
type="hidden" name="jurnal" value="<?php echo
$selectedJurnal; ?>">
                                <input
type="hidden" name="keyword" value="<?php echo $keyword;
?>">
                                <button
type="submit">Go</button>
                                </form>
                                </div>

                                <script>

document.getElementById('startYearFilter').addEventListener

```

```

('change', function() {
    var
    selectedStartYear = this.value;
    var
    selectedEndYear =
    document.getElementById('endYearFilter').value;
    var
    selectedJurnal =
    document.getElementById('jurnalFilter').value;
    var keyword =
    document.getElementById('keywordSearch').value;
    var params =
    new URLSearchParams(window.location.search);
    if
    (selectedStartYear !== 'none') {
    params.set('start_year', selectedStartYear);
    } else {

    params.delete('start_year');
    }
    if
    (selectedEndYear !== 'none') {
    params.set('end_year', selectedEndYear);
    } else {

    params.delete('end_year');
    }
    if
    (selectedJurnal !== 'none') {
    params.set('jurnal', selectedJurnal);
    } else {

    params.delete('jurnal');
    }
    if (keyword
    !== '') {
    params.set('keyword', keyword);
    } else {

    params.delete('keyword');
    }

    window.location.href = 'Halaman_Artikel.php?' +
    params.toString();
    });

```

```

document.getElementById('endYearFilter').addEventListener
('change', function() {
    var
    selectedStartYear =
    document.getElementById('startYearFilter').value;
    var
    selectedEndYear = this.value;
    var
    selectedJurnal =
    document.getElementById('jurnalFilter').value;
    var keyword =
    document.getElementById('keywordSearch').value;
    var params =
    new URLSearchParams(window.location.search);
    if
    (selectedStartYear !== 'none') {
    params.set('start_year', selectedStartYear);
    } else {
    params.delete('start_year');
    }
    if
    (selectedEndYear !== 'none') {
    params.set('end_year', selectedEndYear);
    } else {
    params.delete('end_year');
    }
    if
    (selectedJurnal !== 'none') {
    params.set('jurnal', selectedJurnal);
    } else {
    params.delete('jurnal');
    }
    if (keyword
    !== '') {
    params.set('keyword', keyword);
    } else {
    params.delete('keyword');
    }

    window.location.href = 'Halaman_Artikel.php?' +
    params.toString();
    });

```

```

                                                                    if (keyword
!= ' ') {
params.set('keyword', keyword);
                                                                    } else {

params.delete('keyword');
                                                                    }

window.location.href = 'Halaman_Artikel.php?' +
params.toString();
                                                                    });

document.getElementById('jurnalFilter').addEventListener(
'change', function() {
                                                                    var
selectedStartYear =
document.getElementById('startYearFilter').value;
                                                                    var
selectedEndYear =
document.getElementById('endYearFilter').value;
                                                                    var
selectedJurnal = this.value;
                                                                    var keyword =
document.getElementById('keywordSearch').value;
                                                                    var params =
new URLSearchParams(window.location.search);
                                                                    if
(selectedStartYear !== 'none') {
params.set('start_year', selectedStartYear);
                                                                    } else {

params.delete('start_year');
                                                                    }
                                                                    if
(selectedEndYear !== 'none') {
params.set('end_year', selectedEndYear);
                                                                    } else {

params.delete('end_year');
                                                                    }
                                                                    if
(selectedJurnal !== 'none') {
params.set('jurnal', selectedJurnal);
                                                                    } else {

```

```

params.delete('jurnal');
                                }
                                if (keyword
    !== '') {
    params.set('keyword', keyword);
                                } else {
    params.delete('keyword');
                                }

    window.location.href = 'Halaman_Artikel.php?' +
    params.toString();
                                });

    document.getElementById('keywordSearch').addEventListener
    ('keypress', function(e) {
                                if (e.key ===
    'Enter') {
    e.preventDefault();

                                var
    selectedStartYear =
    document.getElementById('startYearFilter').value;
                                var
    selectedEndYear =
    document.getElementById('endYearFilter').value;
                                var
    selectedJurnal =
    document.getElementById('jurnalFilter').value;
                                var
    keyword = this.value;
                                var
    params = new URLSearchParams(window.location.search);
                                if
    (selectedStartYear !== 'none') {
    params.set('start_year', selectedStartYear);
                                } else {
    params.delete('start_year');
                                }
                                if
    (selectedEndYear !== 'none') {
    params.set('end_year', selectedEndYear);
                                } else {
    params.delete('end_year');
                                }

```

```

params.delete('jurnal');

                                                                    if
(selectedJurnal !== 'none') {

params.set('jurnal', selectedJurnal);

                                                                    } else {

params.delete('jurnal');

                                                                    }
                                                                    if
(keyword !== '') {

params.set('keyword', keyword);

                                                                    } else {

params.delete('keyword');

                                                                    }

window.location.href = 'Halaman_Artikel.php?' +
params.toString();

                                                                    }
                                                                    });

document.getElementById('jump_page_form').addEventListener
('submit', function(e) {

e.preventDefault();

                                                                    var jumpPage
= document.getElementById('jump_page').value;
                                                                    var
selectedStartYear =
document.getElementById('startYearFilter').value;
                                                                    var
selectedEndYear =
document.getElementById('endYearFilter').value;
                                                                    var
selectedJurnal =
document.getElementById('jurnalFilter').value;
                                                                    var keyword =
document.getElementById('keywordSearch').value;
                                                                    var params =
new URLSearchParams(window.location.search);

params.set('page', jumpPage);

                                                                    if
(selectedStartYear !== 'none') {

params.set('start_year', selectedStartYear);

                                                                    } else {

```

```

params.delete('start_year');

                                                                    }
                                                                    if
(selectedEndYear !== 'none') {
params.set('end_year', selectedEndYear);
                                                                    } else {
params.delete('end_year');
                                                                    }
                                                                    if
(selectedJurnal !== 'none') {
params.set('jurnal', selectedJurnal);
                                                                    } else {
params.delete('jurnal');
                                                                    }
                                                                    if (keyword
!== '') {
params.set('keyword', keyword);
                                                                    } else {
params.delete('keyword');
                                                                    }

window.location.href = 'Halaman_Artikel.php?' +
params.toString();
                                                                    });

document.getElementById('clear_filters').addEventListener
('click', function(e) {
                                                                    var params =
new URLSearchParams(window.location.search);
params.delete('start_year');
params.delete('end_year');
params.delete('jurnal');
params.delete('keyword');

window.location.href = 'Halaman_Artikel.php?' +
params.toString();
                                                                    });
</script>

```

```

        </div>
    </div> <!-- /.card -->
</div> <!-- /.col-lg-12 -->
</div>
</div>
<!-- /.orders -->
</div>
<!-- .animated -->
</div>

<!-- /.content -->
<div class="clearfix"></div>
<!-- Footer -->
<footer class="site-footer">
    <div class="footer-inner bg-white">
        <div class="row">
            <div class="col-sm-6">
                Copyright &copy; 2024 Hasyim
Yahya
            </div>
        </div>
    </div>
</footer>
<!-- /.site-footer -->
</div>
<!-- /#right-panel -->

<!-- Scripts -->
<script
src="https://cdn.jsdelivr.net/npm/jquery@2.2.4/dist/jquery.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.4/dist/umd/popper.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/jquery-match-height@0.7.2/dist/jquery.matchHeight.min.js"></script>
<script src="assets/js/main.js"></script>

<!-- Chart js -->
<script
src="https://cdn.jsdelivr.net/npm/chart.js@2.7.3/dist/Chart.bundle.min.js"></script>

<!--Chartist Chart-->
<script
src="https://cdn.jsdelivr.net/npm/chartist@0.11.0/dist/ch

```

```

plugin-legend@0.6.2/chartist-plugin-
legend.min.js"></script>

<script
src="https://cdn.jsdelivrivr.net/npm/jquery.flot@0.8.3/jquery.
flot.min.js"></script>
<script src="https://cdn.jsdelivrivr.net/npm/flot-
pie@1.0.0/src/jquery.flot.pie.min.js"></script>
<script src="https://cdn.jsdelivrivr.net/npm/flot-
spline@0.0.1/js/jquery.flot.spline.min.js"></script>

<script
src="https://cdn.jsdelivrivr.net/npm/simpleweather@3.1.0/jquer
y.simpleWeather.min.js"></script>
<script src="assets/js/init/weather-init.js"></script>

<script
src="https://cdn.jsdelivrivr.net/npm/moment@2.22.2/moment.min.
js"></script>
<script
src="https://cdn.jsdelivrivr.net/npm/fullcalendar@3.9.0/dist/f
ullcalendar.min.js"></script>
<script src="assets/js/init/fullcalendar-
init.js"></script>

</body>
</html>

```

Lampiran 4 Skrip Kode Halaman_Artikel.phph Sesudah penerapan Prinsip Clean Code

```

<?php
include "Koneksi.php";
?>
<body>
    <?php include "left_panel.php"?>
    <!-- /#left-panel -->
    <!-- Right Panel -->
    <div id="right-panel" class="right-panel">
        <!-- Header-->
        <?php include "header.php"?>
        <!-- /#header -->
        <!-- Content -->
        <div class="content">
            <!-- Animated -->
            <div class="animated fadeIn">
                <div class="clearfix"></div>
                <!-- Orders -->
                <div class="orders">
                    <div class="row">
                        <div class="col-lg-12">
                            <!-- Filter -->
                            <div class="card">
                                <div class="card-header">
                                    <h3 class="box-title
text-center">Filter</h3>
                                </div>
                                <div class="card-body">
                                    <div class="form-row">
                                        <div class="form-
group col-md-6">
                                            <label
for="startYearFilter">Start Year:</label>
                                            <select
class="form-control" id="startYearFilter"
name="start_year">
                                                <option
value="none">None</option>
                                            <?php
                                                $yearQuery
= "SELECT DISTINCT YEAR(tanggal_terbit) as year FROM
artikel ORDER BY year DESC";
$yearResult = mysqli_query($koneksi, $yearQuery);
$selectedStartYear = isset($_GET['start_year']) ?
$_GET['start_year'] : 'none';
while ($row = mysqli_fetch_assoc($yearResult)) {
    $selected = ($row['year'] == $selectedStartYear) ?

```

```

        'selected' : '';
        echo '<option value="' . $row['year'] . '" ' .
$selected . '>' . $row['year'] . '</option>';
    }
?>

</select>
</div>
<div class="form-
group col-md-6">

        <label
for="endYearFilter">End Year:</label>

        <select
class="form-control" id="endYearFilter" name="end_year">
            <option
value="none">None</option>
            <?php
$yearResult = mysqli_query($koneksi, $yearQuery); // Reuse
the same query
$selectedEndYear = isset($_GET['end_year']) ?
$_GET['end_year'] : 'none';
while ($row = mysqli_fetch_assoc($yearResult)) {
    $selected = ($row['year'] == $selectedEndYear) ?
    'selected' : '';
    echo '<option value="' . $row['year'] . '" ' .
$selected . '>' . $row['year'] . '</option>';
}
?>

            </select>
        </div>
    </div>
    <div class="form-row">
        <div class="form-
group col-md-4">

            <label
for="sortFilter">Sort By:</label>

            <select
class="form-control" id="sortFilter" name="sort_by">
                <option
value="none" <?php echo isset($_GET['sort_by']) &&
$_GET['sort_by'] === 'none' ? 'selected' : ''; ?>>None
                </option>
                <option
value="nama" <?php echo isset($_GET['sort_by']) &&
$_GET['sort_by'] === 'nama' ? 'selected' : ''; ?>>Nama
                </option>
                <option
value="tanggal_terbit" <?php echo isset($_GET['sort_by'])
&& $_GET['sort_by'] === 'tanggal terbit' ? 'selected' :
''; ?>>Tanggal

```

```

</select>
</div>
<div class="form-
group col-md-4">
<label
for="jurnalFilter">Journal Filter:</label>
<select
class="form-control" id="jurnalFilter" name="jurnal">
<option
value="none">None</option>
<?php
    $jurnalQuery = "SELECT id_jurnal, nama_jurnal FROM jurnal
ORDER BY nama_jurnal ASC";
    $jurnalResult = mysqli_query($koneksi, $jurnalQuery);
    $selectedJurnal = isset($_GET['jurnal']) ? $_GET['jurnal']
: 'none';
    while ($row = mysqli_fetch_assoc($jurnalResult)) {
        $selected = ($row['nama_jurnal'] == $selectedJurnal) ?
'selected' : '';
        echo '<option value="' . $row['nama_jurnal'] . '" ' .
$selected . '>' . $row['nama_jurnal'] . '</option>';
    }
?>
</select>
</div>
<div class="form-
group col-md-4">
<label
for="keywordSearch">Keyword Search:</label>
<input
type="text" class="form-control" id="keywordSearch"
name="keyword"
placeholder="Enter keywords..."
value="<?php echo isset($_GET['keyword']) ?
$_GET['keyword'] : ''; ?>"
</div>
</div>
</div>
</div>
</div> <!-- /.col-lg-12 -->
</div>
<!-- /.orders -->
</div>
<!-- Animated -->
<div class="animated fadeIn">
<div class="clearfix"></div>
<!-- Orders -->
<div class="orders">

```

```

        <div class="row">
            <div class="col-lg-12">
                <div class="card">
                    <div class="card-header">
                        <h3 class="box-title
text-center">Artikel Ilmiah</h3>
                    </div>
                    <div class="card-body">
                        <div class="table-
stats order-table ov-h">
                            <table
class="table" id="table-1">
                                <thead>
                                    <tr>

<th>No</th>

<th>Nama Artikel</th>

<th>Tanggal Terbit</th>

<th>Asal Artikel</th>

                                </tr>
                                </thead>
                                <tbody>
                                    <?php

// Pagination Variables
$limit = 20; // Number of items per page
$page = isset($_GET['page']) && is_numeric($_GET['page'])
? $_GET['page'] : 1; // Current page, default to 1 if not
valid
$start = ($page - 1) * $limit; // Starting index for
fetching data
// Default query to fetch all articles
$query = "SELECT * FROM artikel";
// Handle filters
$conditions = [];
// Start Year filter
if (isset($_GET['start_year']) && $_GET['start_year'] !==
'none') {
    $selectedStartYear =
mysqli_real_escape_string($koneksi, $_GET['start_year']);
    $conditions[] = "YEAR(tanggal_terbit) >=
$_selectedStartYear";
}
// End Year filter
if (isset($_GET['end_year']) && $_GET['end_year'] !==
'none') {
    $selectedEndYear = mysqli_real_escape_string($koneksi,
$_GET['end_year']);
    $conditions[] = "YEAR(tanggal_terbit) <=

```

```

$selectedEndYear";
} elseif (isset($_GET['start_year']) &&
$_GET['start_year'] !== 'none') {
    // If only start year is selected, show articles from
    start year to max year
    $maxYearQuery = "SELECT MAX(YEAR(tanggal_terbit)) as
max_year FROM artikel";
    $maxYearResult = mysqli_query($koneksi,
$maxYearQuery);
    $maxYear =
mysqli_fetch_assoc($maxYearResult)['max_year'];
    $conditions[] = "YEAR(tanggal_terbit) <= $maxYear";
}
// Journal filter
if (isset($_GET['jurnal']) && $_GET['jurnal'] !== 'none')
{
    $selectedJurnal = mysqli_real_escape_string($koneksi,
$_GET['jurnal']);
    $jurnalIdQuery = "SELECT id_jurnal FROM jurnal WHERE
nama_jurnal = '$selectedJurnal'";
    $jurnalIdResult = mysqli_query($koneksi,
$jurnalIdQuery);
    if ($jurnalIdRow =
mysqli_fetch_assoc($jurnalIdResult)) {
        $jurnalId = $jurnalIdRow['id_jurnal'];
        $conditions[] = "asal_artikel = $jurnalId";
    }
}
// Keyword search
$keyword = '';
if (isset($_GET['keyword']) && $_GET['keyword'] !== '') {
    $keyword = mysqli_real_escape_string($koneksi,
$_GET['keyword']);
    $conditions[] = "(nama_artikel LIKE '%$keyword%' OR
penulis_artikel LIKE '%$keyword%')";
}
// Combine conditions
if (!empty($conditions)) {
    $query .= " WHERE " . implode(" AND ", $conditions);
}

// Handle sorting
$sort_by = isset($_GET['sort_by']) ? $_GET['sort_by'] :
'nama'; // Default sort by 'nama'
$sort_order = 'ASC'; // Default order

if ($sort_by === 'tanggal_terbit') {
    $sort_order = 'DESC'; // Sort by 'tanggal_terbit'
    descending
}

```

```

// Add order and limit clause
$query = " ORDER BY ";
if ($sort_by === 'nama') {
    $query = "nama_artikel ASC";
} elseif ($sort_by === 'tanggal_terbit') {
    $query = "tanggal_terbit $sort_order";
}
$query = " LIMIT $start, $limit";
// Fetch data from database
$result = mysqli_query($koneksi, $query);
// Check if the query was successful
if ($result) {
    // Display fetched data
    $no = $start + 1;
    while ($data = mysqli_fetch_array($result)) {
        ?>
<tr>

<td><?php echo $no++; ?>
</td>
<td><a
class="nama_artikel"
href="<?php echo $data['link_artikel']; ?>"><?php echo
$data['nama_artikel']; ?></a>

<br>

<br>

Penulis:

<?php echo $data['penulis_artikel']; ?>
</td>

<td><span
class="tanggal_terbit"><?php echo $data['tanggal_terbit'];
?></span>
</td>
<td>

<span class="asal_artikel">

<?php
        $idtest = $data['asal_artikel'];
        $str = strval($idtest);
        $asal = mysqli_query($koneksi, "SELECT nama_jurnal
FROM jurnal WHERE id_jurnal = $str");
        $asal_artikel = mysqli_fetch_assoc($asal);

```

```

        $asal_final = implode(" ", $asal_artikel);
        echo $asal_final;
    ?>

</span>

</td>
</tr>
<?php
    }
} else {
    echo "<tr><td colspan='4'>No data found</td></tr>";
}
?>

</tbody>
</table>
</div> <!-- /.table-
stats -->
<div class="pagination
justify-content-center">
    <?php
        $total_query =
"SELECT COUNT(*) as total FROM artikel";
if (!empty($conditions)) {
    $total_query .= " WHERE " . implode(" AND ",
$conditions);
}
$total_result = mysqli_query($koneksi, $total_query);
if ($total_result) {
    $total_data =
mysqli_fetch_assoc($total_result)['total'];
    $total_pages = ceil($total_data / $limit);
    // Previous page link
    if ($page > 1) {
        echo '<a class="page-link" href="?page=' . ($page
- 1) . '&start_year=' . $selectedStartYear . '&end_year='
. $selectedEndYear . '&jurnal=' . $selectedJurnal .
'&keyword=' . $keyword . '">Previous</a>';
    }
    $start_page = max(1, $page - 2); // Adjust starting
page based on current page
    $end_page = min($total_pages, $page + 2); // Adjust
ending page based on
    for ($i = $start_page; $i < $page; $i++) {
        $active_class = ($page == $i) ? ' active' : '';
        echo '<a class="page-link" . $active_class . '"
href="?page=' . $i . '&start_year=' . $selectedStartYear .
'&end_year=' . $selectedEndYear . '&jurnal=' .
$selectedJurnal . '&keyword=' . $keyword . '"> . $i .
'</a>';
    }
    // Display current page link

```

```

        echo '<a class="page-link active" href="?page=' .
$page . '&start_year=' . $selectedStartYear . '&end_year='
. $selectedEndYear . '&jurnal=' . $selectedJurnal .
'&keyword=' . $keyword . '>' . $page . '</a>';
    // Display last 2 page links (if applicable)
    if ($end_page > $page) {
        for ($i = $page + 1; $i <= $end_page; $i++) {
            $active_class = ($page == $i) ? ' active' :
'';
            echo '<a class="page-link" . $active_class .
' href="?page=' . $i . '&start_year=' .
$selectedStartYear . '&end_year=' . $selectedEndYear .
'&jurnal=' . $selectedJurnal . '&keyword=' . $keyword .
'>' . $i . '</a>';
        }
        // Next page link
        if ($page < $total_pages) {
            echo '<a class="page-link" href="?page=' . ($page
+ 1) . '&start_year=' . $selectedStartYear . '&end_year='
. $selectedEndYear . '&jurnal=' . $selectedJurnal .
'&keyword=' . $keyword . '>Next</a>';
        }
    }
?>

</div>
<div id="jump-to-page"
style="display: always; margin-top: 5px;">
    <form action=""
method="get">
        <label
for="jump_page">Jump to page:</label>
        <input
type="number" id="jump_page" name="page" min="1"
max="<?php
echo $total_pages; ?>"
value="1">
        <input
type="hidden" name="start_year"
value="<?php echo $selectedStartYear; ?>"
        <input
type="hidden" name="end_year"
value="<?php echo $selectedEndYear; ?>"
        <input
type="hidden" name="jurnal"
value="<?php echo $selectedJurnal; ?>"
        <input
type="hidden" name="keyword"

```

```

value="<?php echo $keyword; ?>">
                                <button
type="submit">Go</button>
                                </form>
                                </div>
                                </div>
                                </div> <!-- /.card -->
                                </div> <!-- /.col-lg-12 -->
                                </div>
                                </div>
                                <!-- /.orders -->
                                </div>
                                <!-- .animated -->
                                </div>
                                <!-- /.content -->
                                <div class="clearfix"></div>
                                <!-- Footer -->
                                <?php include "footer.php"?>
                                <script
src="./assets/js/scriptArtikel.js"></script>
                                </body>

                                </html>

```

Lampiran 5 Skrip Kode indeks.php Sebelum penerapan Prinsip Clean Code

```
<!doctype html>
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8
lt-ie7" lang=""> <![endif]-->
<!--[if IE 7]>         <html class="no-js lt-ie9 lt-ie8"
lang=""> <![endif]-->
<!--[if IE 8]>         <html class="no-js lt-ie9" lang="">
<![endif]-->
<!--[if gt IE 8]><!-->
<html class="no-js" lang=""> <!--<![endif]-->

<head>
  <meta charset="utf-8">
  <!-- <meta http-equiv="refresh" content="180"> -->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Halaman Home</title>
  <meta name="description" content="Monitoring Jurnal">
  <meta name="viewport" content="width=device-width,
initial-scale=1">

  <link rel="apple-touch-icon" href="images/uinws.png">
  <link rel="shortcut icon" href="images/uinws.png">

  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/normalize.css@8.0.0/nor
malize.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/bootstrap@4.1.3/dist/cs
s/bootstrap.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/font-
awesome@4.7.0/css/font-awesome.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/gh/lykmapipo/themify-
icons@0.1.2/css/themify-icons.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/pixeden-stroke-7-
icon@1.2.3/pe-icon-7-stroke/dist/pe-icon-7-
stroke.min.css">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/flag-icon-
css/3.2.0/css/flag-icon.min.css">
  <link rel="stylesheet" href="assets/css/cs-skin-
elastic.css">
  <link rel="stylesheet" href="assets/css/style.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jq
```

```

.min.js"></script>
<!-- <script type="text/javascript"
src="https://cdn.jsdelivr.net/html5shiv/3.7.3/html5shiv.mi
n.js"></script> -->
<link
href="https://cdn.jsdelivr.net/npm/chartist@0.11.0/dist/ch
artist.min.css" rel="stylesheet">
<link
href="https://cdn.jsdelivr.net/npm/jqvmap@1.5.1/dist/jqvma
p.min.css" rel="stylesheet">

<link
href="https://cdn.jsdelivr.net/npm/weathericons@2.1.0/css/
weather-icons.css" rel="stylesheet" />
<link
href="https://cdn.jsdelivr.net/npm/fullcalendar@3.9.0/dist
/fullcalendar.min.css" rel="stylesheet" />

<script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/1.5.3/js
pdf.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf-
autotable/3.5.6/jspdf.plugin.autotable.min.js"></script>
<script type="text/javascript"
src="assets/js/pdf/main.js"></script>

</head>

<body>
<!-- Left Panel -->
<aside id="left-panel" class="left-panel">
<nav class="navbar navbar-expand-sm navbar-default">
<div id="main-menu" class="main-menu collapse
navbar-collapse">
<ul class="nav navbar-nav">
<li class="">
<a href="Halaman_Home.php"><i class="menu-icon
fa fa-home"></i>Halaman Home </a>
</li>
<li class="">
<a href="Halaman_Jurnal.php"><i class="menu-
icon fa fa-laptop"></i>Halaman Jurnal</a>
</li>
<li class="">
<a href="Halaman_Artikel.php"><i class="menu-
icon fa fa-book"></i>Halaman Artikel</a>
</li>
</ul>
</div><!-- /.navbar-collapse -->
</nav>
</body>

```

```

</aside>
<!-- /#left-panel -->
<!-- Right Panel -->
<div id="right-panel" class="right-panel">
  <!-- Header-->
  <header id="header" class="header">
    <div class="top-left">
      <div class="navbar-header">
        <a class="navbar-brand" href=""></a>
        <a class="navbar-brand hidden" href=""></a>
        <a id="menuToggle" class="menutoggle"><i
class="fa fa-bars"></i></a>
      </div>
    </div>
  </header>
  <!-- /#header -->
  <!-- Content -->
  <div class="content">
    <div class="row">
      <div class="col-md-12">
        <div class="card">
          <div class="card-header">
            <h3 class="box-title text-
center">Informations</h3>
          </div>
          <div class="card-body">
            <p class="card-text">Informasi Mengenai
Sistem Monitoring:<br>
            <br>
            1. Script Web Scraper untuk mengecek update
jurnal dijalankan secara otomatis setiap jam<br>
            2. Tombol Check Update pada halaman Home
digunakan untuk memeriksa pembaruan secara mandiri oleh
pengguna<br>
            3. Tombol Check Update akan menjalankan
Script Web Scraper di latar belakang sistem (durasi
sekitar 5-10 menit)<br>
            4. Proses Check Update akan tetap berjalan
bahkan jika halaman website ditutup<br>
            5. Proses Check Update yang masih berjalan
ditandai dengan simbol refresh yang berjalan<br>
            6. Progress Bar fungsi Check Update akan
penuh apabila fungsi Check Update selesai menjalankan
Script Web Scraper dan halaman Home tidak ditutup<br>
            7. Progress Bar fungsi Check Update tidak
akan terisi apabila mengganti halaman Home dengan halaman
lain<br>
            8. Data artikel ditampilkan secaraurut
sesuai nama alfabet<br>
            9. Tombol Check Update akan menjalankan

```

9. Apabila data artikel tidak tampil secara otomatis ketika fungsi filter digunakan, harap kembali ke page 1 (jump to page = 1)

10. Apabila terdapat "Fatal Error" ketika menampilkan data artikel, hal ini berarti sistem sedang menjalankan Script Web Scraper. Mohon refresh Halaman Artikel ketika hal ini terjadi.

```

</p><br>
<br>
<?php
    if (isset($_POST['checkUpdate'])) {
        $descriptorspec = array(
            0 => array("pipe", "r"), // stdin
            1 => array("pipe", "w"), // stdout
            2 => array("pipe", "w")  // stderr
        );

        $process =
proc_open('/home/monito29/virtualenv/public_html/jurnalscr
aper/3.10/bin/python3 JurnalScraper.py &',
        $descriptorspec, $pipes);

        if (is_resource($process)) {
            fclose($pipes[0]);

            $output =
stream_get_contents($pipes[1]);
            fclose($pipes[1]);

            $error =
stream_get_contents($pipes[2]);
            fclose($pipes[2]);
        }
    }
?>
<div class="progress mb-3">
<?php
    if (isset($_POST['checkUpdate'])) {
        echo '<div class="progress-bar bg-
success" role="progressbar" style="width: 100%" aria-
valuenow="100" aria-valuemin="0" aria-
valuemax="100"></div>';
    } else {
        echo '<div class="progress-bar bg-
success" role="progressbar" style="width: 0%" aria-
valuenow="0" aria-valuemin="0" aria-
valuemax="100"></div>';
    }
?>
</div>

```

```

        <div class="row mt-4">
            <div class="col-md-12 d-flex justify-
content-center">
                <form action="" method="post">
                    <button type="submit" class="btn
button-78" name="checkUpdate" id="checkUpdateBtn">Check
Update</button>
                </form>
            </div>
        </div>
    </div>
</div>
<!-- /.content -->

<div class="clearfix"></div>
<!-- Footer -->
<footer class="site-footer">
    <div class="footer-inner bg-white">
        <div class="row">
            <div class="col-sm-6">
                Copyright &copy; 2024 Hasyim Yahya
            </div>
        </div>
    </div>
</div>
<!-- /.site-footer -->
</div>
<!-- /#right-panel -->

<!-- Scripts -->
<script
src="https://cdn.jsdelivr.net/npm/jquery@2.2.4/dist/jquery
.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.4/dist/um
d/popper.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/
bootstrap.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/jquery-match-
height@0.7.2/dist/jquery.matchHeight.min.js"></script>
<script src="assets/js/main.js"></script>

<!-- Chart js -->
<script
src="https://cdn.jsdelivr.net/npm/chart.js@2.7.3/dist/Char
t.bundle.min.js"></script>

```

```

    <!--Chartist Chart-->
    <script
src="https://cdn.jsdelivr.net/npm/chartist@0.11.0/dist/chartist.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/chartist-plugin-legend@0.6.2/chartist-plugin-legend.min.js"></script>

    <script
src="https://cdn.jsdelivr.net/npm/jquery.flot@0.8.3/jquery.flot.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/flot-pie@1.0.0/src/jquery.flot.pie.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/flot-spline@0.0.1/js/jquery.flot.spline.min.js"></script>

    <script
src="https://cdn.jsdelivr.net/npm/simpleweather@3.1.0/jquery.simpleWeather.min.js"></script>
    <script src="assets/js/init/weather-init.js"></script>

    <script
src="https://cdn.jsdelivr.net/npm/moment@2.22.2/moment.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/fullcalendar@3.9.0/dist/fullcalendar.min.js"></script>
    <script src="assets/js/init/fullcalendar-init.js"></script>

</body>

</html>

```

Lampiran 6 Skrip indeks.php Sesudah Penerapan Prinsip Clean Code

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Halaman home</title>
  <meta name="description" content="Monitoring Jurnal">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
</head>

<body>
  <?php include "left_panel.php"?>
  <div id="right-panel" class="right-panel">
    <?php include "header.php"?>
    <div class="content">
      <div class="row">
        <div class="col-md-12">
          <div class="card">
            <div class="card-header">
              <h3 class="box-title text-
center">Information</h3>
            </div>
            <div class="card-body">
              <p class="card-text">Information
regarding the journal and article monitoring system.</p>
              <div class="progress mb-3">
                <div class="progress-bar bg-
success" role="progressbar"
                    style="width: <?=
isset($_POST['checkUpdate']) ? '100%' : '0%' ?>"
                    aria-valuenow="<?=
isset($_POST['checkUpdate']) ? '100' : '0' ?>"
                    aria-valuemin="0" aria-
valuemax="100"></div>
              </div>
              <div class="row mt-4">
                <div class="col-md-12 d-flex
justify-content-center">
                  <form action=""
method="post">
                    <button
type="submit" class="btn button-78" name="checkUpdate">Check
Update</button>
```

```

        </form>
    </div>
</div>
<?php
    if
(isset($_POST['checkUpdate'])) {
        runUpdateScript();
    }

    function runUpdateScript()
    {
        $descriptorspec = [
            0 => ["pipe", "r"],
            1 => ["pipe", "w"],
            2 => ["pipe", "w"]
        ];
        $process = proc_open('/path/to/python3 script.py &',
$descriptorspec, $pipes);

        if (is_resource($process)) {
            fclose($pipes[0]);
            $output = stream_get_contents($pipes[1]);
            fclose($pipes[1]);
            fclose($pipes[2]);
        }
    }
?>

</div>
</div>
</div>
</div>
<?php include "footer.php"?>
</div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js"></script>
</body>

</html>

```

Lampiran 7 Skrip kode Halaman_Jurnal.php Sebelum penerapan Prinsip Clean Code

```
<!doctype html>
<!--[if lt IE 7]>          <html class="no-js lt-ie9 lt-ie8
lt-ie7" lang=""> <![endif]-->
<!--[if IE 7]>            <html class="no-js lt-ie9 lt-ie8"
lang=""> <![endif]-->
<!--[if IE 8]>            <html class="no-js lt-ie9" lang="">
<![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js" lang=""> <!--
<![endif]-->
<head>
  <meta charset="utf-8">
  <!-- <meta http-equiv="refresh" content="180"> -->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Halaman Jurnal</title>
  <meta name="description" content="Monitoring Jurnal">
  <meta name="viewport" content="width=device-width,
initial-scale=1">

  <link rel="apple-touch-icon" href="images/uinws.png">
  <link rel="shortcut icon" href="images/uinws.png">

  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/normalize.css@8.0.0/nor
malize.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/cs
s/bootstrap.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/font-
awesome@4.7.0/css/font-awesome.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/gh/lykmapipo/themify-
icons@0.1.2/css/themify-icons.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/pixeden-stroke-7-
icon@1.2.3/pe-icon-7-stroke/dist/pe-icon-7-
stroke.min.css">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/flag-icon-
css/3.2.0/css/flag-icon.min.css">
  <link rel="stylesheet" href="assets/css/cs-skin-
elastic.css">
  <link rel="stylesheet" href="assets/css/style.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jq
```

```

    <!-- <script type="text/javascript"
src="https://cdn.jsdelivr.net/html5shiv/3.7.3/html5shiv.mi
n.js"></script> -->
    <link
href="https://cdn.jsdelivr.net/npm/chartist@0.11.0/dist/ch
artist.min.css" rel="stylesheet">
    <link
href="https://cdn.jsdelivr.net/npm/jqvmap@1.5.1/dist/jqvma
p.min.css" rel="stylesheet">

    <link
href="https://cdn.jsdelivr.net/npm/weathericons@2.1.0/css/
weather-icons.css" rel="stylesheet" />
    <link
href="https://cdn.jsdelivr.net/npm/fullcalendar@3.9.0/dist
/fullcalendar.min.css" rel="stylesheet" />

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/1.5.3/js
pdf.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf-
autotable/3.5.6/jspdf.plugin.autotable.min.js"></script>
    <script type="text/javascript"
src="assets/js/pdf/main.js"></script>

</head>

<body>
    <!-- Left Panel -->
    <aside id="left-panel" class="left-panel">
        <nav class="navbar navbar-expand-sm navbar-
default">
            <div id="main-menu" class="main-menu collapse
navbar-collapse">
                <ul class="nav navbar-nav">
                    <li class="">
                        <a href="Halaman_Home.php"><i
class="menu-icon fa fa-home"></i>Halaman Home </a>
                    </li>
                    <li class="">
                        <a href="Halaman_Jurnal.php"><i
class="menu-icon fa fa-laptop"></i>Halaman Jurnal</a>
                    </li>
                    <li class="">
                        <a href="Halaman_Artikel.php"><i
class="menu-icon fa fa-book"></i>Halaman Artikel</a>
                    </li>
                </ul>
            </div><!-- /.navbar-collapse -->
        </nav>
    </aside>

```

```

<!-- /#left-panel -->
<!-- Right Panel -->
<div id="right-panel" class="right-panel">
  <!-- Header-->
    <header id="header" class="header">
      <div class="top-left">
        <div class="navbar-header">
          <a class="navbar-brand" href=""></a>
          <a class="navbar-brand hidden"
href=""></a>
          <a id="menuToggle"
class="menutoggle"><i class="fa fa-bars"></i></a>
        </div>
      </div>
    </header>
    <!-- /#header -->
    <!-- Content -->
    <div class="content">
      <!-- Animated -->
      <div class="animated fadeIn">
        <div class="clearfix"></div>
        <!-- Orders -->
        <div class="orders">
          <div class="row">
            <div class="col-lg-12">
              <!-- Filter -->
              <div class="card">
                <div class="card-header">
                  <h3 class="box-title
text-center">Filter</h3>
                </div>
                <div class="card-body">
                  <form id="filterForm">
                    <div class="form-
row">
                      <div
class="form-group col-md-6">
                        <label
for="sortingFilter">Sort by:</label>
                        <select
class="form-control" id="sortingFilter" name="sorting">
<option value="none" <?php echo (!isset($_GET['sorting'])
|| $_GET['sorting'] == 'none') ? 'selected' : '';
?>>None</option>
<option value="name" <?php echo (isset($_GET['sorting'])
&& $_GET['sorting'] == 'name') ? 'selected' : '';
?>>Nama</option>

```

```

<option value="indeks" <?php echo (isset($_GET['sorting'])
&& $_GET['sorting'] == 'indeks') ? 'selected' : '';
?>>Indeks</option>

<option value="volume" <?php echo (isset($_GET['sorting'])
&& $_GET['sorting'] == 'volume') ? 'selected' : '';
?>>Volume Terbaru</option>

</select>
</div>
<div
class="form-group col-md-6">
<label
for="keywordSearch">Journal Search:</label>
<input
type="text" class="form-control" id="keywordSearch"
name="keyword" placeholder="Enter Journal Name..."
value="<?php echo isset($_GET['keyword']) ?
htmlspecialchars($_GET['keyword']) : ''; ?>"
/>
</div>
</form>
</div>
</div>
</div> <!-- /.col-lg-12 -->
</div>
</div>
<!-- /.orders -->
</div>

<!-- JavaScript for automatic filtering and
searching -->
<script>

document.getElementById('sortingFilter').addEventListener(
'change', function() {
    updateQueryParams();
});

document.getElementById('keywordSearch').addEventListener(
'keydown', function(e) {
    if (e.key === 'Enter') {
        e.preventDefault();
        updateQueryParams();
    }
});

function updateQueryParams() {
    const sorting =
document.getElementById('sortingFilter').value;
    const keyword =

```

```

document.getElementById('keywordSearch').value;
    let queryParams = '?page=1';

    if (sorting !== 'none') {
        queryParams += '&sorting=' +
sorting;
    }
    if (keyword) {
        queryParams += '&keyword=' +
encodeURIComponent(keyword);
    }

    window.location.href =
window.location.pathname + queryParams;
}
</script>

<!-- Animated -->
<div class="animated fadeIn">
    <div class="clearfix"></div>
    <!-- Orders -->
    <div class="orders">
        <div class="row">
            <div class="col-lg-12">
                <div class="card">
                    <div class="card-header">
                        <h3 class="box-title
text-center">Jurnal Ilmiah</h3>
                    </div>
                    <div class="card-body">
                        <div class="table-stats
order-table ov-h">
                            <table class="table"
id="table-1">
                                <thead>
                                    <tr>
                                        <th>No</th>
                                        <th>Nama
Jurnal</th>
                                        <th>Indeks SINTA & Jadwal Terbit</th>
                                        <th>Volume Terbaru</th>
                                    </tr>
                                </thead>
                                <tbody>
                                    <?php
include
"Koneksi.php";

```

```

// Get query parameters for sorting and search
                                                                    $sorting =
isset($_GET['sorting']) ? $_GET['sorting'] : 'none';
                                                                    $keyword =
isset($_GET['keyword']) ?
mysqli_real_escape_string($koneksi, $_GET['keyword']) : '';

                                                                    //
Pagination Variables
                                                                    $limit = 20;
// Number of items per page
                                                                    $page =
isset($_GET['page']) ? intval($_GET['page']) : 1; // Current
page
                                                                    $start =
($page - 1) * $limit; // Starting index for fetching data

                                                                    // Build the
base query
                                                                    $query =
"SELECT * FROM jurnal WHERE 1";

                                                                    // Add
keyword search filter
                                                                    if ($keyword
!= ' ') {
                                                                    $query
.= " AND nama_jurnal LIKE '%$keyword%'";
                                                                    }

                                                                    // Add
sorting logic
                                                                    switch
($sorting) {
                                                                    case
'name':

$query .= " ORDER BY nama_jurnal ASC";

break;
                                                                    case
'indeks':

$query .= " ORDER BY FIELD(indeks_jurnal, 'S1', 'S2', 'S3',
'S4', 'S5', 'S6', 'Not Found')";

break;
                                                                    case
'volume':

$query .= " ORDER BY (SELECT MAX(tanggal_terbit) FROM

```

```

artikel WHERE asal_artikel = jurnal.id_jurnal) DESC";

break;
                                default:

$query .= " ORDER BY nama_jurnal ASC";
                                }

                                // Add
pagination limit
                                $query .= "
LIMIT $start, $limit";

                                $result =
mysqli_query($koneksi, $query);

                                // Display
fetched data
                                $no = $start
+ 1;
                                while ($data
= mysqli_fetch_array($result)) {

    $volume_query = "SELECT MAX(tanggal_terbit) as max_date FROM
artikel WHERE asal_artikel = " . intval($data['id_jurnal']);

    $volume_result = mysqli_query($koneksi, $volume_query);

    $volume_data = mysqli_fetch_assoc($volume_result);

    $tanggal_final = $volume_data['max_date'];

                                ?>
                                <tr>

<td><?php echo $no++; ?></td>

<td><a class="nama_jurnal" href="<?php echo
$data['link_jurnal']; ?>"><?php echo $data['nama_jurnal'];
?></a></td>

<td><a class="indeks_jurnal" href="<?php echo
$data['link_indeks']; ?>"><?php echo $data['indeks_jurnal'];
?></a>

<br>

<?php echo $data['jadwal_terbit']; ?>

</td>

                                <td>

```

```

<span class="volume_terbaru">

<?php echo $data['volume_terbaru']; ?><br>

Diterbitkan pada:

<?php echo $tanggal_final; ?>

</span>

</td>

</tr>
<?php

}
?>
</tbody>
</table>
</div> <!-- /.table-

stats -->

<!-- Pagination Links --
>

<div class="pagination
justify-content-center">

<?php
$total_query =
"SELECT COUNT(*) as total FROM jurnal WHERE 1";

// Add keyword
search filter to count query
if ($keyword !== '')
{
$total_query .=
" AND nama_jurnal LIKE '%$keyword%'";
}

$total_result =
mysqli_query($koneksi, $total_query);
$total_data =
mysqli_fetch_assoc($total_result) ['total'];
$total_pages =
ceil($total_data / $limit);

// Previous page
link
if ($page > 1) {
echo '<a
class="page-link" href="?page=' . ($page - 1) . '&sorting='
. $sorting . '&keyword=' . urlencode($keyword) .
">Previous</a>';
}

```

```

// Page links
for ($i = 1; $i <=
$total_pages; $i++) {
    $active_class =
($page == $i) ? ' active' : '';
    echo '<a
class="page-link" . $active_class . "' href="?page=' . $i .
'&sorting=' . $sorting . '&keyword=' . urlencode($keyword) .
'">' . $i . '</a>';
}

// Next page link
if ($page <
$total_pages) {
    echo '<a
class="page-link" href="?page=' . ($page + 1) . '&sorting='
. $sorting . '&keyword=' . urlencode($keyword) .
'">Next</a>';
}
?>
</div>
</div>
</div> <!-- /.card -->
</div> <!-- /.col-lg-12 -->
</div>
</div>
<!-- /.orders -->
</div>
<!-- .animated -->
</div>

<!-- /.content -->
<div class="clearfix"></div>
<!-- Footer -->
<footer class="site-footer">
    <div class="footer-inner bg-white">
        <div class="row">
            <div class="col-sm-6">
                Copyright &copy; 2024 Hasyim Yahya
            </div>
        </div>
    </div>
</div>
<!-- /.site-footer -->
</div>
<!-- /#right-panel -->

<!-- Scripts -->
<script

```

```

src="https://cdn.jsdelivr.net/npm/jquery@2.2.4/dist/jquery.m
in.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.4/dist/umd/
popper.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bo
otstrap.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/jquery-match-
height@0.7.2/dist/jquery.matchHeight.min.js"></script>
<script src="assets/js/main.js"></script>

<!-- Chart js -->
<script
src="https://cdn.jsdelivr.net/npm/chart.js@2.7.3/dist/Chart.
bundle.min.js"></script>

<!--Chartist Chart-->
<script
src="https://cdn.jsdelivr.net/npm/chartist@0.11.0/dist/chart
ist.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/chartist-
plugin-legend@0.6.2/chartist-plugin-legend.min.js"></script>

<script
src="https://cdn.jsdelivr.net/npm/jquery.flot@0.8.3/jquery.f
lot.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/flot-
pie@1.0.0/src/jquery.flot.pie.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/flot-
spline@0.0.1/js/jquery.flot.spline.min.js"></script>

<script
src="https://cdn.jsdelivr.net/npm/simpleweather@3.1.0/jquery
.simpleWeather.min.js"></script>
<script src="assets/js/init/weather-init.js"></script>

<script
src="https://cdn.jsdelivr.net/npm/moment@2.22.2/moment.min.j
s"></script>
<script
src="https://cdn.jsdelivr.net/npm/fullcalendar@3.9.0/dist/fu
llcalendar.min.js"></script>
<script src="assets/js/init/fullcalendar-
init.js"></script>

</body>
</html>

```

Lampiran 8 Skrip kode Halaman_Jurnal.php Sesudah penerapan Prinsip Clean Code

```
<!doctype html>
<html class="no-js" lang="">

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Halaman Jurnal</title>
  <meta name="description" content="Monitoring Jurnal">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
</head>

<body>
  <!-- Left Panel -->
  <?php include "left_panel.php"?>
  <!-- /#left-panel -->
  <!-- Right Panel -->
  <div id="right-panel" class="right-panel">
    <!-- Header-->
    <?php include "header.php"?>
    <!-- /#header -->
    <!-- Content -->
    <div class="content">
      <!-- Animated -->
      <div class="animated fadeIn">
        <div class="clearfix"></div>
        <!-- Orders -->
        <div class="orders">
          <div class="row">
            <div class="col-lg-12">
              <!-- Filter -->
              <div class="card">
                <div class="card-header">
                  <h3 class="box-title
text-center">Filter</h3>
                </div>
                <div class="card-body">
                  <form id="filterForm">
                    <div class="form-
row">
                      <div
class="form-group col-md-6">
                        <label
for="sortingFilter">Sort by:</label>
                        <select
```

```

<label for="sortingFilter">Sort by:</label>
                                <select
class="form-control" id="sortingFilter" name="sorting">
                                <option
value="none" <?php echo (!isset($_GET['sorting']) ||
$_GET['sorting'] == 'none') ? 'selected' : ''; ?>>None

                                </option>
                                <option
value="name" <?php echo (isset($_GET['sorting']) &&
$_GET['sorting'] == 'name') ? 'selected' : ''; ?>>Nama

                                </option>
                                <option
value="indeks" <?php echo (isset($_GET['sorting']) &&
$_GET['sorting'] == 'indeks') ? 'selected' : ''; ?>>Indeks

                                </option>
                                <option
value="volume" <?php echo (isset($_GET['sorting']) &&
$_GET['sorting'] == 'volume') ? 'selected' : ''; ?>>Volume

Terbaru</option>
                                </select>
                                </div>
                                <div
class="form-group col-md-6">
                                <label
for="keywordSearch">Journal Search:</label>
                                <input
type="text" class="form-control" id="keywordSearch"
name="keyword" placeholder="Enter Journal Name..."
value="<?php echo isset($_GET['keyword']) ?
htmlspecialchars($_GET['keyword']) : ''; ?>"
                                </div>
                                </div>
                                </form>
                                </div>
                                </div>
                                </div> <!-- /.col-lg-12 -->
                                </div>
                                </div>
                                <!-- /.orders -->
                                </div>
                                <!-- Animated -->
                                <div class="animated fadeIn">
                                <div class="clearfix"></div>
                                <!-- Orders -->
                                <div class="orders">
                                <div class="row">

```

```

        <div class="row">
            <div class="col-lg-12">
                <div class="card">
                    <div class="card-header">
                        <h3 class="box-title
text-center">Jurnal Ilmiah</h3>
                    </div>
                    <div class="card-body">
                        <div class="table-stats
order-table ov-h">
                            <table class="table"
id="table-1">
                                <thead>
                                    <tr>
                                        <th>No</th>
                                        <th>Nama
Jurnal</th>
                                        <th>Indeks SINTA & Jadwal Terbit</th>
                                        <th>Volume Terbaru</th>
                                    </tr>
                                </thead>
                                <tbody>
                                    <?php
include
"Koneksi.php";
// Get query
parameters for sorting and search
                                $sorting =
isset($_GET['sorting']) ? $_GET['sorting'] : 'none';
                                $keyword =
isset($_GET['keyword']) ?
mysqli_real_escape_string($koneksi, $_GET['keyword']) : '';
                                //
Pagination Variables
                                $limit = 20;
// Number of items per page
                                $page =
isset($_GET['page']) ? intval($_GET['page']) : 1; // Current
page
                                $start =
($page - 1) * $limit; // Starting index for fetching data
                                // Build the
base query
                                $query =
"SELECT * FROM jurnal WHERE 1";
                                // Add
keyword search filter
                                if ($keyword
!= '' ) {

```

```

!= ' ') {
                                                                    $query
.= " AND nama_jurnal LIKE '%$keyword%'";
                                                                    }
                                                                    // Add
sorting logic
                                                                    switch
($sorting) {
                                                                    case
'name':

$query .= " ORDER BY nama_jurnal ASC";

break;
                                                                    case
'indeks':

$query .= " ORDER BY FIELD(indeks_jurnal, 'S1', 'S2', 'S3',
'S4', 'S5', 'S6', 'Not Found')";

break;
                                                                    case
'volume':

$query .= " ORDER BY (SELECT MAX(tanggal_terbit) FROM
artikel WHERE asal_artikel = jurnal.id_jurnal) DESC";

break;
                                                                    default:

$query .= " ORDER BY nama_jurnal ASC";
                                                                    }
                                                                    // Add
pagination limit
                                                                    $query .= "
LIMIT $start, $limit";
                                                                    $result =
mysqli_query($koneksi, $query);
                                                                    // Display
fetched data
                                                                    $no = $start
+ 1;
                                                                    while ($data
= mysqli_fetch_array($result)) {

$volume_query = "SELECT MAX(tanggal terbit) as max date FROM
artikel WHERE asal_artikel = " . intval($data['id_jurnal']);

$volume_result = mysqli_query($koneksi, $volume_query);

$volume_data = mysqli_fetch_assoc($volume_result);

```

```

$tanggal_final = $volume_data['max_date'];
                                ?>
                                <tr>

<td><?php echo $no++; ?>
                                </td>
                                <td><a
class="nama_jurnal"
href="<?php echo $data['link_jurnal']; ?>"><?php echo
$data['nama_jurnal']; ?></a>
                                </td>
                                <td><a
class="indeks_jurnal"
href="<?php echo $data['link_indeks']; ?>"><?php echo
$data['indeks_jurnal']; ?></a>
                                <br>

<?php echo $data['jadwal_terbit']; ?>
                                </td>
                                <td>

<span class="volume_terbaru">

<?php echo $data['volume_terbaru']; ?><br>
Diterbitkan pada:

<?php echo $tanggal_final; ?>

</span>
                                </td>
                                </tr>
                                <?php
    }
    ?>
                                </tbody>
                                </table>
                                </div> <!-- /.table-
stats -->
                                <!-- Pagination Links --
>
                                <div class="pagination
justify-content-center">
                                <?php
                                    $total_query =
"SELECT COUNT(*) as total FROM jurnal WHERE 1";
                                    // Add keyword
search filter to count query
                                    if ($keyword != '')

```

```

        if ($keyword != '')
    {
        $total_query .=
" AND nama_jurnal LIKE '%$keyword%';"
    }
    $total_result =
mysqli_query($koneksi, $total_query);
    $total_data =
mysqli_fetch_assoc($total_result)['total'];
    $total_pages =
ceil($total_data / $limit);

    // Previous page
    link
    if ($page > 1) {
        echo '<a
class="page-link" href="?page=' . ($page - 1) . '&sorting='
. $sorting . '&keyword=' . urlencode($keyword) .
">Previous</a>';
    }
    // Page links
    for ($i = 1; $i <=
$total_pages; $i++) {
        $active_class =
($page == $i) ? ' active' : '';
        echo '<a
class="page-link" . $active_class . " href="?page=' . $i .
'&sorting=' . $sorting . '&keyword=' . urlencode($keyword) .
">' . $i . '</a>';
    }
    // Next page link
    if ($page <
$total_pages) {
        echo '<a
class="page-link" href="?page=' . ($page + 1) . '&sorting='
. $sorting . '&keyword=' . urlencode($keyword) .
">Next</a>';
    }
    ?>
</div>
</div>
</div> <!-- /.card -->
</div> <!-- /.col-lg-12 -->
</div>
</div>
<!-- /.orders -->
</div>
<!-- .animated -->
</div>
<!-- /.content -->
<div class="clearfix"></div>
<!-- Footer -->
<?php include "footer.php"?>

```

```
        <!-- Footer -->
        <?php include "footer.php"?>
        <!-- /.site-footer -->
    </div>
    <!-- /#right-panel -->
    <script src="./assets/js/scriptJurnal.js"></script>
</body>

</html>
```

Lampiran 9 Skrip kode Halaman_Home.php Sebelum penerapan Prinsip Clean Code

```
<!doctype html>
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8 lt-
ie7" lang=""> <![endif]-->
<!--[if IE 7]>         <html class="no-js lt-ie9 lt-ie8"
lang=""> <![endif]-->
<!--[if IE 8]>         <html class="no-js lt-ie9" lang="">
<![endif]-->
<!--[if gt IE 8]><!-->
<html class="no-js" lang=""> <!--<![endif]-->

<head>
  <meta charset="utf-8">
  <!-- <meta http-equiv="refresh" content="180"> -->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Halaman Home</title>
  <meta name="description" content="Monitoring Jurnal">
  <meta name="viewport" content="width=device-width, initial-
scale=1">

  <link rel="apple-touch-icon" href="images/uinws.png">
  <link rel="shortcut icon" href="images/uinws.png">

  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/normalize.css@8.0.0/normal
ize.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/bootstrap@4.1.3/dist/css/b
ootstrap.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/font-
awesome@4.7.0/css/font-awesome.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/gh/lykmapipo/themify-
icons@0.1.2/css/themify-icons.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/pixeden-stroke-7-
icon@1.2.3/pe-icon-7-stroke/dist/pe-icon-7-stroke.min.css">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/flag-icon-
css/3.2.0/css/flag-icon.min.css">
  <link rel="stylesheet" href="assets/css/cs-skin-
elastic.css">
  <link rel="stylesheet" href="assets/css/style.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquer
y.min.js"></script>
  <!-- <script type="text/javascript">
```

```

src="https://cdn.jsdelivr.net/html5shiv/3.7.3/html5shiv.min.js
"></script> -->
    <link
href="https://cdn.jsdelivr.net/npm/chartist@0.11.0/dist/charti
st.min.css" rel="stylesheet">
    <link
href="https://cdn.jsdelivr.net/npm/jqvmap@1.5.1/dist/jqvmap.mi
n.css" rel="stylesheet">

    <link
href="https://cdn.jsdelivr.net/npm/weathericons@2.1.0/css/weat
her-icons.css" rel="stylesheet" />
    <link
href="https://cdn.jsdelivr.net/npm/fullcalendar@3.9.0/dist/ful
lcalendar.min.css" rel="stylesheet" />

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/1.5.3/jspdf.
min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf-
autotable/3.5.6/jspdf.plugin.autotable.min.js"></script>
    <script type="text/javascript"
src="assets/js/pdf/main.js"></script>

</head>

<body>
    <!-- Left Panel -->
    <aside id="left-panel" class="left-panel">
        <nav class="navbar navbar-expand-sm navbar-default">
            <div id="main-menu" class="main-menu collapse navbar-
collapse">
                <ul class="nav navbar-nav">
                    <li class="">
                        <a href="Halaman_Home.php"><i class="menu-icon fa
fa-home"></i>Halaman Home </a>
                    </li>
                    <li class="">
                        <a href="Halaman_Jurnal.php"><i class="menu-icon
fa fa-laptop"></i>Halaman Jurnal</a>
                    </li>
                    <li class="">
                        <a href="Halaman_Artikel.php"><i class="menu-icon
fa fa-book"></i>Halaman Artikel</a>
                    </li>
                </ul>
            </div><!-- /.navbar-collapse -->
        </nav>
    </aside>
    <!-- #left-panel -->
    <!-- Right Panel -->

```

```

<div id="right-panel" class="right-panel">
  <!-- Header-->
  <header id="header" class="header">
    <div class="top-left">
      <div class="navbar-header">
        <a class="navbar-brand" href=""></a>
        <a class="navbar-brand hidden" href=""></a>
        <a id="menuToggle" class="menutoggle"><i class="fa
fa-bars"></i></a>
      </div>
    </div>
  </header>
  <!-- /#header -->
  <!-- Content -->
  <div class="content">
    <div class="row">
      <div class="col-md-12">
        <div class="card">
          <div class="card-header">
            <h3 class="box-title text-
center">Informations</h3>
          </div>
          <div class="card-body">
            <p class="card-text">Informasi Mengenai Sistem
Monitoring:<br>
            <br>
            1. Script Web Scraper untuk mengecek update
jurnal dijalankan secara otomatis setiap jam<br>
            2. Tombol Check Update pada halaman Home
digunakan untuk memeriksa pembaruan secara mandiri oleh
pengguna<br>
            3. Tombol Check Update akan menjalankan Script
Web Scraper di latar belakang sistem (durasi sekitar 5-10
menit)<br>
            4. Proses Check Update akan tetap berjalan
bahkan jika halaman website ditutup<br>
            5. Proses Check Update yang masih berjalan
ditandai dengan simbol refresh yang berjalan<br>
            6. Progress Bar fungsi Check Update akan penuh
apabila fungsi Check Update selesai menjalankan Script Web
Scraper dan halaman Home tidak ditutup<br>
            7. Progress Bar fungsi Check Update tidak akan
terisi apabila mengganti halaman Home dengan halaman lain<br>
            8. Data artikel ditampilkan secaraurut sesuai
nama alfabet<br>
            9. Apabila data artikel tidak tampil secara
otomatis ketika fungsi filter digunakan, harap kembali ke page
1 (jump to page = 1)<br>
            10. Apabila terdapat "Fatal Error" ketika

```

menampilkan data artikel, hal ini berarti sistem sedang menjalankan Script Web Scraper. Mohon refresh Halaman Artikel ketika hal ini terjadi.

```
</p><br>
<br>
<?php
    if (isset($_POST['checkUpdate'])) {
        $descriptorspec = array(
            0 => array("pipe", "r"), // stdin
            1 => array("pipe", "w"), // stdout
            2 => array("pipe", "w") // stderr
        );

        $process =
proc_open('/home/monito29/virtualenv/public_html/jurnalscraper/3
.10/bin/python3 JurnalScraper.py &', $descriptorspec, $pipes);

        if (is_resource($process)) {
            fclose($pipes[0]);

            $output = stream_get_contents($pipes[1]);
            fclose($pipes[1]);

            $error = stream_get_contents($pipes[2]);
            fclose($pipes[2]);
        }
    }
?>
<div class="progress mb-3">
<?php
    if (isset($_POST['checkUpdate'])) {
        echo '<div class="progress-bar bg-success"
role="progressbar" style="width: 100%" aria-valuenow="100" aria-
valuemin="0" aria-valuemax="100"></div>';
    } else {
        echo '<div class="progress-bar bg-success"
role="progressbar" style="width: 0%" aria-valuenow="0" aria-
valuemin="0" aria-valuemax="100"></div>';
    }
?>
</div>

<div class="row mt-4">
    <div class="col-md-12 d-flex justify-
content-center">
        <form action="" method="post">
            <button type="submit" class="btn button-
78" name="checkUpdate" id="checkUpdateBtn">Check Update</button>
        </form>
    </div>
</div>
```

```

        </div>
    </div>
</div>
</div>
</div>
</div>
<!-- /.content -->

<div class="clearfix"></div>
<!-- Footer -->
<footer class="site-footer">
    <div class="footer-inner bg-white">
        <div class="row">
            <div class="col-sm-6">
                Copyright &copy; 2024 Hasyim Yahya
            </div>
        </div>
    </div>
</div>
<!-- /.site-footer -->
</div>
<!-- #right-panel -->

<!-- Scripts -->
<script
src="https://cdn.jsdelivr.net/npm/jquery@2.2.4/dist/jquery.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.4/dist/umd/popper.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/jquery-match-height@0.7.2/dist/jquery.matchHeight.min.js"></script>
<script src="assets/js/main.js"></script>

<!-- Chart js -->
<script
src="https://cdn.jsdelivr.net/npm/chart.js@2.7.3/dist/Chart.bundle.min.js"></script>

<!--Chartist Chart-->
<script
src="https://cdn.jsdelivr.net/npm/chartist@0.11.0/dist/chartist.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/chartist-plugin-legend@0.6.2/chartist-plugin-legend.min.js"></script>

<script

```

```

src="https://cdn.jsdelivr.net/npm/jquery.flot@0.8.3/jquery.flot.
min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/flot-
pie@1.0.0/src/jquery.flot.pie.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/flot-
spline@0.0.1/js/jquery.flot.spline.min.js"></script>

  <script
src="https://cdn.jsdelivr.net/npm/simpleweather@3.1.0/jquery.sim
pleWeather.min.js"></script>
  <script src="assets/js/init/weather-init.js"></script>

  <script
src="https://cdn.jsdelivr.net/npm/moment@2.22.2/moment.min.js"><
/script>
  <script
src="https://cdn.jsdelivr.net/npm/fullcalendar@3.9.0/dist/fullca
lendar.min.js"></script>
  <script src="assets/js/init/fullcalendar-init.js"></script>

</body>

</html>

```

Lampiran 10 Skrip Kode Halaman_Home.php sesidah penerapan prinsip Clean Code

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Halaman home</title>
  <meta name="description" content="Monitoring Jurnal">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
</head>

<body>
  <?php include "left_panel.php"?>
  <div id="right-panel" class="right-panel">
    <?php include "header.php"?>
    <div class="content">
      <div class="row">
        <div class="col-md-12">
          <div class="card">
            <div class="card-header">
              <h3 class="box-title text-
center">Information</h3>
            </div>
            <div class="card-body">
              <p class="card-text">Information about the
Monitoring System:<br><br>
                1. The Web Scraper script checks for
journal updates automatically every hour.<br>
                2. The Check Update button on the Home
page allows users to manually check for updates.<br>
                3. The Check Update button runs the Web
Scraper script in the background (duration approximately
5-10
                    minutes).<br>
                4. The Check Update process continues even
if the website page is closed.<br>
                5. The ongoing Check Update process is
indicated by a running refresh symbol.<br>
                6. The Check Update progress bar will fill
up when the script completes and the Home page remains
open.<br>
                7. The Check Update progress bar will not
fill if the Home page is replaced with another page.<br>
                8. Article data is displayed in
alphabetical order.<br>
                9. If article data does not display
```

9. If article data does not display automatically when the filter function is used, please return to

page 1.

10. If a "Fatal Error" occurs while displaying article data, it means the system is running the Web

Scraper script. Please refresh the Article Page when this happens.

</p>

<?php

```
function runUpdateScript()
{
    $descriptors = [
        0 => ["pipe", "r"],
// stdin
        1 => ["pipe", "w"],
// stdout
        2 => ["pipe", "w"]
// stderr
    ];

    $pythonProcess =
proc_open('/home/monito29/virtualenv/public_html/jurnalscr
aper/3.10/bin/python3 JurnalScraper.py &', $descriptors,
$processPipes);

    if
(is_resource($pythonProcess)) {
fclose($processPipes[0]);

        $scriptOutput =
stream_get_contents($processPipes[1]);
fclose($processPipes[1]);

        $scriptError =
stream_get_contents($processPipes[2]);
fclose($processPipes[2]);
    }
}

if (isset($_POST['checkUpdate'])) {
    runUpdateScript();
}
?>

<div class="progress mb-3">
<?php
```

```

        $progressWidth =
isset($_POST['checkUpdate']) ? '100' : '0';
        echo "<div class='progress-bar bg-success'
role='progressbar' style='width: {$progressWidth}%' aria-
valuenow='{$progressWidth}' aria-valuemin='0' aria-
valuemax='100'></div>";
        ?>
    </div>

    <div class="row mt-4">
        <div class="col-md-12 d-flex justify-
content-center">
            <form action="" method="post">
                <button type="submit" class="btn
button-78" name="checkUpdate" id="checkUpdateBtn">Check
                Update</button>
            </form>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="clearfix"></div>
<?php include "footer.php"; ?>
</div>
</body>

</html>

```

Lampiran 11 Skrip Kode IndexScraper.py Sebelum penerapan Prinsip Clean Code

```
<?php
#!/usr/bin/env python
# coding: utf-8

# In[ ]:

import requests
from bs4 import BeautifulSoup
import mysql.connector
import datetime
import time # for rate limiting (optional)

def scrape_journals_uin_walisongo(search_url,
indeks_jurnal_id):
    """
    Scrapes the accreditation of journals listed on the
    SINTA search page
    that are affiliated with UIN Walisongo Semarang.

    Args:
        search_url (str): The URL of the search page on
        SINTA website.
        indeks_jurnal_id (int): The ID of the journal in
        the database.

    Returns:
        str: The extracted accreditation (or None if not
        found).
    """

    response = requests.get(search_url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Find all journal listings
    journal_listings = soup.find_all('div', class_='list-
item row mt-3')

    journals = []
    for listing in journal_listings:
        # Check affiliation location (more generic
        selector)
        affiliation_element = listing.find('div',
class_=lambda class_name: class_name and 'affil-loc' in
class name and 'mt-2' in class name)
```

```

        if affiliation_element and "Walisongo" in
affiliation_element.text:
            # Extract journal name
            journal_name_element = listing.find('div',
class_='affil-name mb-3')
            if journal_name_element:
                journal_name =
journal_name_element.find('a').text.strip()

            # Find accreditation element (enhanced
selector)
            accreditation_element = listing.find('span',
class_=lambda class_name: class_name and 'num-stat' in
class_name and 'accredited' in class_name)
            accreditation = None
            if accreditation_element:
                accreditation =
accreditation_element.text.strip().split()[0]

            # Add journal details to dictionary
            journal_info = {
                "accreditation": accreditation
            }
            journals.append(journal_info)
    return journals

def scrape_sinta_value(search_url, indeks_jurnal_id):
    """
    Scrapes the value of the `` tag with the text
    "Sinta 2" from the provided URL.

    Args:
        search_url (str): The URL of the journal profile page
        on SINTA website.
        indeks_jurnal_id (int): The ID of the journal in the
        database.

    Returns:
        str: The extracted Sinta value with "Sinta " replaced
        by "S" (or None if not found).
    """

    response = requests.get(search_url)
    response.raise_for_status() # Raise an exception for
failed requests

    soup = BeautifulSoup(response.content, "html.parser")

    # Find the card containing "Sinta 2" text
    sinta_card = soup.find("div". class ="card bg-card-

```

```

        # Extract the text from the `<stat num>` tag within the
        card (if it exists)
        sinta_value = None
        if sinta_card:
            sinta_value = sinta_card.find("div", class_="stat-
num").text

        # Modify Sinta value if found
        if sinta_value:
            sinta_value = sinta_value.split()[1] # Get the
second word (number)
            sinta_value = "S" + sinta_value # Prepend "S"

        return sinta_value

def scrape_index(search_urls):
    """
    Iterates through a list of search URLs and attempts to
    scrape journals using
    scrape_journals_uin_walisongo first, then
    scrape_sinta_value if the first fails.

    Args:
        search_urls (list): A list of URLs to scrape.

    Returns:
        None
    """

    for i, url in enumerate(search_urls):
        print(f"Link no.{i+1}: ", end='')

        # Fetch journal ID using afetchone()
        sql_get_id = "SELECT id_jurnal FROM jurnal WHERE
link_indeks = (%s)"
        data = (url,) # Wrap base_url in a tuple for the
query parameter
        cursor.execute(sql_get_id, data)
        indeks_jurnal_id = cursor.fetchone()[0] # Extract
the first column (id_jurnal) from the result

        journals = scrape_journals_uin_walisongo(url,
indeks_jurnal_id)
        if journals:
            for journal in journals:
                if journal['accreditation']:
                    # Store accreditation in database
                    sql = "UPDATE jurnal SET indeks_jurnal
= %s WHERE id_jurnal = %s"
                    data = (journal['accreditation'],

```

```

        data = (journal['accreditation'],
indeks_jurnal_id) # Use extracted value
        cursor.execute(sql, data)
        db.commit()

        print(f"Accreditation:
{journal['accreditation']}", "id: ", indeks_jurnal_id)
    else:
        sinta_value = scrape_sinta_value(url,
indeks_jurnal_id)
        if sinta_value:
            # Store Sinta value in database
            sql = "UPDATE jurnal SET indeks_jurnal = %s
WHERE id_jurnal = %s"
            data = (sinta_value, indeks_jurnal_id)
            cursor.execute(sql, data)
            db.commit()

            print(f"Sinta value: {sinta_value}", "id:
", indeks_jurnal_id)
        else:
            # Store accreditation in database
            sql = "UPDATE jurnal SET indeks_jurnal = %s
WHERE id_jurnal = %s"
            not_found = "Not Found"
            data = (not_found, indeks_jurnal_id)
            cursor.execute(sql, data)
            db.commit()

            print("Index not found.", "id: ",
indeks_jurnal_id)

# Connect to database
db = mysql.connector.connect(
    host="localhost",
    username="monito29_admin",
    password="monito29_admin",
    database="monito29_db",
    charset= "utf8"
)

cursor = db.cursor()

# Retrieve search URLs from the database
cursor.execute("SELECT link_indeks FROM jurnal")
search_urls = [x[0] for x in cursor.fetchall()]

# Iterate through search URLs and scrape data
scrape_index(search_urls)

# Close database connection
db.close()

```

Lampiran 12 Skrip Kode IndexScraper.py sesudah penerapan Prinsip Clean Code

```
import requests
from bs4 import BeautifulSoup
import mysql.connector

def fetch_page_content(url):
    """Fetches and returns the HTML content of a given
    URL."""
    response = requests.get(url)
    response.raise_for_status()
    return BeautifulSoup(response.content, "html.parser")

def scrape_journal_accreditation(search_url):
    """Scrapes accreditation data for journals affiliated
    with UIN Walisongo from SINTA."""
    soup = fetch_page_content(search_url)
    journals = []

    for listing in soup.find_all("div", class_="list-item
row mt-3"):
        affiliation = listing.find("div", class_=lambda
x: x and "affil-loc" in x)
        if affiliation and "Walisongo" in
affiliation.text:
            journal_name = listing.find("div",
class_="affil-name mb-3")
            journal_name =
journal_name.find("a").text.strip() if journal_name else
None

            accreditation = listing.find(
                "span", class_=lambda x: x and
"accredited" in x
            )
            accreditation = (
                accreditation.text.strip().split()[0] if
accreditation else None
            )

            journals.append({"name": journal_name,
"accreditation": accreditation})

    return journals
```

```

def scrape_sinta_index(search_url):
    """Scrapes the Sinta index value from a journal
    profile page."""
    soup = fetch_page_content(search_url)
    sinta_card = soup.find("div", class_="card bg-card-
orange")

    if sinta_card:
        sinta_value = sinta_card.find("div",
class_="stat-num").text.split()[1]
        return f"S{sinta_value}"
    return None

def update_journal_accreditation(cursor, journal_id,
accreditation):
    """Updates the journal accreditation in the
    database."""
    cursor.execute(
        "UPDATE jurnal SET indeks_jurnal = %s WHERE
id_jurnal = %s",
        (accreditation, journal_id),
    )

def process_journals(cursor, journals):
    """Processes journals and updates their accreditation
    in the database."""
    for idx, (journal_id, url) in enumerate(journals,
start=1):
        print(f"Processing journal {idx}: ", end="")

        journals_data = scrape_journal_accreditation(url)
        if journals_data:
            for journal in journals_data:
                if journal["accreditation"]:
                    update_journal_accreditation(
                        cursor, journal_id,
journal["accreditation"])
                )
            print(
                f"Accreditation:
{journals_data[0]['accreditation']}, ID: {journal_id}"
            )
        else:

```

```

        sinta_value = scrape_sinta_index(url)
        update_journal_accreditation(cursor,
journal_id, sinta_value or "Not Found")
        print(
            f"Accreditation: {sinta_value if
sinta_value else 'Not Found'}, ID: {journal_id}"
        )

def main():
    """Establishes database connection and processes
journals."""
    db_config = {
        "host": "localhost",
        "username": "root",
        "password": "InitPass12@",
        "database": "jurnal_monitoring_db_baru",
        "charset": "utf8",
    }

    with mysql.connector.connect(**db_config) as db:
        cursor = db.cursor()
        cursor.execute("SELECT id_jurnal, link_indeks
FROM jurnal")
        process_journals(cursor, cursor.fetchall())
        db.commit()

if __name__ == "__main__":
    main()

```

Lampiran 13 Skrip JurnalScraper.py Sebelum menerapkan Prinsip Clean Code

```
#!/usr/bin/env python
# coding: utf-8

# In[ ]:

#FINAL CODE FOR CHECK UPDATE
#jangan lupa ganti balik db koneksi dari localhost ke
monito29

#FINAL CODE FOR CHECK UPDATE
#jangan lupa ganti balik db koneksi dari localhost ke
monito29

import requests
from bs4 import BeautifulSoup
import mysql.connector
import datetime
from threading import Thread # for multithreading
import time # for rate limiting (optional)
import re

start_time = time.time()

def joint_seminar():
    base_url =
    "https://journal.walisongo.ac.id/index.php/PJIS/issue/arch
ive"
    cursor.execute("INSERT INTO `jurnal` (`id_jurnal`,
`nama_jurnal`, `link_jurnal`, `jadwal_terbit`,
`archive_jurnal`, `volume_terbaru`, `link_indeks`,
`indeks_jurnal`) VALUES ('30', 'Joint International
Seminar',
'https://journal.walisongo.ac.id/index.php/PJIS',
'Unknown',
'https://journal.walisongo.ac.id/index.php/PJIS/issue/arch
ive', NULL,
'https://sinta.kemdikbud.go.id/journals/index/?q=Joint+Int
ernational+Seminar', 'Not Found');")
    db.commit()
    delete_query = """
        DELETE a
        FROM artikel AS a
        INNER JOIN (
            SELECT link_artikel, MIN(id_artikel) AS min_id
            FROM artikel
            GROUP BY link_artikel
```

```

        ) AS b ON a.link_artikel = b.link_artikel AND
a.id_artikel <> b.min_id
"""

cursor.execute(delete_query)
db.commit()

page = requests.get(base_url)
soup = BeautifulSoup(page.content, "html.parser")

# Find all issue links (assuming they are within
anchor tags)
issue_element = soup.find("div",
class_=("issueDescription", "issueDescriptionImage",
"issueCoverImage"))
issue_link = issue_element.find("a")["href"]
latest_volume = "2015"

# Update volume terbaru in jurnal table for the
current archive_jurnal
update_query = """
UPDATE jurnal
SET volume_terbaru = %s
WHERE archive_jurnal = %s
"""
cursor.execute(update_query, (latest_volume,
base_url))
db.commit()

print(f"Latest volume for {base_url}:
{latest_volume}") # Confirmation message

archive_jurnal_id = int(30)

try:
    page_article = requests.get(issue_link)
    soup2 = BeautifulSoup(page_article.content,
"html.parser")

    # Try scraping from TOC image link first
    toc_image_element = soup2.find("div",
id="issueCoverImage")
    if toc_image_element and
toc_image_element.find("a", href=True):
        toc_image_link =
        toc_image_element.find("a")["href"]
        toc_page =requests.get(toc_image_link)
        soup3 = BeautifulSoup(toc_page.content,
"html.parser")

        # Get current date
        current date = datetime.date.today()

```

```

        # Format the date as YYYY-MM-DD
        formatted_date = current_date.strftime('%Y-%m-
%d')

        penulis_artikel = "Unknown"
        artikel_parent = soup3.find_all("div",
class_="tocTitle")

        for detail in artikel_parent:
            nama_artikel = (
                detail.find("a", href=True).text
                if detail.find("a", href=True)
                else "Front Matter"
            )
            link_artikel = (
                detail.find("a", href=True)["href"]
                if detail.find("a", href=True)
                else "#"
            )

            insert_query = """
                INSERT INTO artikel (nama_artikel,
asal_artikel, link_artikel, tanggal_terbit,
penulis_artikel)
                VALUES (%s, %s, %s, %s, %s)
            """
            cursor.execute(insert_query, (nama_artikel,
archive_jurnal_id, link_artikel, formatted_date,
penulis_artikel))
            db.commit()

            print(f"{nama_artikel} {archive_jurnal_id}
{link_artikel} {formatted_date} {penulis_artikel}")

        num_articles = len(artikel_parent)

        print("")
        print(f"Number of articles in this issue:
{num_articles}")
        print("-" * 50) # Separator between issues

    except requests.exceptions.RequestException as e:
        print(f"Error fetching issue link: {e}")
        pass

def RJPTD():
    base url =
"https://journal.walisongo.ac.id/index.php/rjtpd/issue/arc
hive"
    cursor.execute("INSERT INTO `jurnal` (`id jurnal`,

```

```

`nama_jurnal`, `link_jurnal`, `jadwal_terbit`,
`archive_jurnal`, `volume_terbaru`, `link_indeks`,
`indeks_jurnal`) VALUES ('45', 'Research Journal on
Teacher Professional Development',
'https://journal.walisongo.ac.id/index.php/rjtpd', 'Twice
a year',
'https://journal.walisongo.ac.id/index.php/rjtpd/issue/arc
hive', NULL,
'https://sinta.kemdikbud.go.id/journals/index/?q=Research+
Journal+on+Teacher+Professional+Development', 'Not
Found');")
db.commit()

delete_query = """
DELETE a
FROM artikel AS a
INNER JOIN (
    SELECT link_artikel, MIN(id_artikel) AS min_id
    FROM artikel
    GROUP BY link_artikel
) AS b ON a.link_artikel = b.link_artikel AND
a.id_artikel <> b.min_id
"""

cursor.execute(delete_query)
db.commit()

# Fetch the current page
page = requests.get(base_url)
soup = BeautifulSoup(page.content, "html.parser")

# Find all issue links (assuming they are within
anchor tags)
issue_element = soup.find("div", style ="clear:left;")
issue_link = issue_element.find("a")["href"]
latest_volume = issue_element.find("a").text

# Update volume_terbaru in jurnal table for the
current archive_jurnal
update_query = """
UPDATE jurnal
SET volume_terbaru = %s
WHERE archive_jurnal = %s
"""
cursor.execute(update_query, (latest_volume,
base_url))
db.commit()

print(f"Latest volume for {base_url}:
{latest_volume}") # Confirmation message

archive_jurnal_id = int(45)

```

```

try:
    page_article = requests.get(issue_link)
    soup2 = BeautifulSoup(page_article.content,
"html.parser")
    # Fallback: scrape from current issue link

    # Get current date
    current_date = datetime.date.today()

    # Format the date as YYYY-MM-DD
    formatted_date = current_date.strftime('%Y-%m-%d')

    artikel_parent = soup2.find_all("td",
class_="tocArticleTitleAuthors")

    for detail in artikel_parent:
        # Extract article title
        title_element = detail.find("div",
class_="tocTitle")
        title_link = title_element.find("a") if
title_element else None
        nama_artikel = title_link.text.strip() if
title_link else "Front Matter"
        link_artikel = title_link["href"] if
title_link else "#"

        authors = []
        author_elements = detail.find_all("div",
class_="tocAuthors")
        for author_element in author_elements:
            author_text = author_element.text.strip()
            author_name = re.sub(r"\s+", " ",
author_text) # Replace multiple whitespaces with single
space

            formatted_author = f"{author_name}."
            authors.append(formatted_author)

        # Combine and format author names
        penulis_artikel = "\n".join(authors)

        insert_query = """
            INSERT INTO artikel (nama_artikel,
asal_artikel, link_artikel, tanggal_terbit,
penulis_artikel)
            VALUES (%s, %s, %s, %s, %s)
        """
        cursor.execute(insert_query, (nama_artikel,
archive_jurnal_id, link_artikel, formatted_date,
penulis_artikel))
        db.commit()

```

```

        print(f"{nama_artikel} {archive_jurnal_id}
{link_artikel} {formatted_date} {penulis_artikel}")

    num_articles = len(artikel_parent)

    print("")
    print(f"Number of articles in this issue:
{num_articles}")
    print("-" * 50) # Separator between issues

    except requests.exceptions.RequestException as e:
        print(f"Error fetching issue link: {e}")
        pass

def scrape_issue(issue_link, archive_jurnal_id):
    try:
        page_article = requests.get(issue_link)
        soup2 = BeautifulSoup(page_article.content,
"html.parser")

        # Try scraping from TOC image link first
        toc_image_element = soup2.find("div",
id="issueCoverImage", class_="col-sm-3")
        if toc_image_element and
toc_image_element.find("a", href=True):
            toc_image_link =
toc_image_element.find("a")["href"]

        # Combine requests for fetching articles and
TOC details
        toc_page_response =
requests.get(toc_image_link)
        soup3 =
BeautifulSoup(toc_page_response.content, "html.parser")

        tanggal_terbit = soup3.find("div",
class_="infoissue").text.replace(
            " Published: ", ""
        )

        date =
datetime.datetime.strptime(tanggal_terbit, '%Y-%m-
%d').date() # Format date

        artikel_parent = soup3.find all("td",
class_="tocArticleTitleAuthors")
        num_articles = len(artikel_parent)

        for detail in artikel_parent:
            # Extract article title
            title_element = detail.find("div".

```

```

class_="tocTitle")
    title_link = title_element.find("a") if
title_element else None
    nama_artikel = title_link.text.strip() if
title_link else "Front Matter"
    link_artikel = title_link["href"] if
title_link else "#"

    # Extract and format author names (logic
remains the same)
    authors = []
    author_elements = detail.find_all("div",
id="authorString")
    for author_element in author_elements:
        author_text =
author_element.text.strip()
        author_name = re.sub(r"\s+", " ",
author_text) # Replace multiple whitespaces with single
space
        formatted_author = f"{author_name}."
        authors.append(formatted_author)

    # Combine and format author names
    penulis_artikel = "\n".join(authors)

    # Insert data into artikel table
    insert_query = """
        INSERT INTO artikel (nama_artikel,
asal_artikel, link_artikel, tanggal_terbit,
penulis_artikel)
        VALUES (%s, %s, %s, %s, %s)
    """
    cursor.execute(insert_query,
(nama_artikel, archive_jurnal_id, link_artikel,
date,penulis_artikel))
    db.commit()
    print(f"{nama_artikel} {archive_jurnal_id}
{link_artikel} {date} {penulis_artikel}")

    else:
        # Fallback: scrape from current issue link
        tanggal_terbit = soup2.find("div",
class_="infoissue").text.replace(
            " Published: ", ""
        )
        date =
datetime.datetime.strptime(tanggal_terbit, '%Y-%m-
%d').date() # Format date

        artikel_parent = soup2.find_all("td",
class_="tocArticleTitleAuthors")
        num_articles = len(artikel_parent)

```

```

        for detail in artikel_parent:
            # Extract article title
            title_element = detail.find("div",
class_="tocTitle")
            title_link = title_element.find("a") if
title_element else None
            nama_artikel = title_link.text.strip() if
title_link else "Front Matter"
            link_artikel = title_link["href"] if
title_link else "#"

            authors = []
            author_elements = detail.find_all("div",
id="authorString")
            for author_element in author_elements:
                author_text =
author_element.text.strip()
                author_name = re.sub(r"\s+", " ",
author_text) # Replace multiple whitespaces with single
space
                formatted_author = f"{author_name}."
                authors.append(formatted_author)

            # Combine and format author names
            penulis_artikel = "\n".join(authors)

            # Insert data into artikel table
            insert_query = """
INSERT INTO artikel (nama_artikel,
asal_artikel, link_artikel, tanggal_terbit,
penulis_artikel)
VALUES (%s, %s, %s, %s, %s)
"""
            cursor.execute(insert_query,
(nama_artikel, archive_jurnal_id, link_artikel,
date,penulis_artikel))
            db.commit()
            print(f"{nama_artikel} {archive_jurnal_id}
{link_artikel} {date} {penulis_artikel}")

        print("")
        print(f"Number of articles in this issue:
{len(artikel_parent)}")
        print("-" * 50) # Separator between issues

    except requests.exceptions.RequestException as e:
        print(f"Error fetching issue link: {e}")
        pass

def Jurnal(archive_links):

```

```

def Jurnal(archive_links):
    link_number = 1
    for i, base_url in enumerate(archive_links):
        page = requests.get(base_url)
        soup = BeautifulSoup(page.content, "html.parser")

        # Find all issue links (assuming they are within
        anchor tags)
        issue_element = soup.find("div",
class_=("issueDescription", "issueDescriptionImage"))
        if not issue_element:
            print(f"Issue element not found for
{base_url}, skipping.")
            continue

        issue_link = issue_element.find("a")["href"]

        # Extract latest volume (assuming it's in the text
        of the anchor tag)
        latest_volume =
        issue_element.find("a").text.replace("book ", "")

        # Update volume_terbaru in jurnal table for the
        current archive_jurnal
        update_query = """
            UPDATE jurnal
            SET volume_terbaru = %s
            WHERE archive_jurnal = %s
        """
        cursor.execute(update_query, (latest_volume,
base_url))
        db.commit()

        print(f"Latest volume for {base_url}:
{latest_volume}") # Confirmation message

        # Fetch archive_jurnal_id using afetchone()
        sql_get_id = "SELECT id_jurnal FROM jurnal WHERE
archive_jurnal = (%s)"
        data = (base_url,) # Wrap base_url in a tuple for
        the query parameter
        cursor.execute(sql_get_id, data)
        archive_jurnal_id = cursor.fetchone()[0] #
        Extract the first column (id_jurnal) from the result

        scrape_issue(issue_link, archive_jurnal_id) #
        Pass archive_jurnal_id for proper linking

        link_number += 1

# Connect to database
db = mysql.connector.connect(

```

```

db = mysql.connector.connect (
    host="localhost",
    username="monito29_admin",
    password="monito29_admin",
    database="monito29_db",
    charset= "utf8"
)

cursor = db.cursor()
cursor.execute("DELETE FROM jurnal WHERE archive_jurnal =
'https://journal.walisongo.ac.id/index.php/PJIS/issue/arch
ive'")
cursor.execute("DELETE FROM jurnal WHERE archive_jurnal =
'https://journal.walisongo.ac.id/index.php/rjtpd/issue/arc
hive'")
db.commit()

delete_query = """
DELETE a
FROM artikel AS a
INNER JOIN (
    SELECT link_artikel, MIN(id_artikel) AS min_id
    FROM artikel
    GROUP BY link_artikel
) AS b ON a.link_artikel = b.link_artikel AND
a.id_artikel <> b.min_id
"""

cursor.execute(delete_query)
db.commit()

cursor.execute("SELECT archive_jurnal FROM jurnal")
archive_links = [x[0] for x in cursor.fetchall()]

Jurnal(archive_links)
joint_seminar()
RJPTD()

delete_query = """
DELETE a
FROM artikel AS a
INNER JOIN (
    SELECT link_artikel, MIN(id_artikel) AS min_id
    FROM artikel
    GROUP BY link_artikel
) AS b ON a.link_artikel = b.link_artikel AND
a.id_artikel <> b.min_id
"""

cursor.execute(delete_query)
db.commit()

```

```
print("Duplicate rows deleted from artikel table.")

# Close database connection
db.close()

end_time = time.time()
execution_time = end_time - start_time
print("Execution time:", execution_time)
```

Lampiran 14 Skrip JurnalScrapper.py Sesudah penerapan Prinsip Clean Code

```
import requests
from bs4 import BeautifulSoup
import mysql.connector
import datetime
import time
import re

start_time = time.time()

def connect_to_database():
    return mysql.connector.connect(
        host="localhost",
        username="root",
        password="InitPass12@",
        database="jurnal_monitoring_db_baru",
        charset="utf8",
    )

def execute_query(cursor, query, data=None, many=False):
    if data:
        if many:
            cursor.executemany(query, data)
        else:
            cursor.execute(query, data)
    else:
        cursor.execute(query)
    cursor._connection.commit()

def insert_articles(cursor, articles, issue_url):
    if not articles:
        print(f"No articles found in {issue_url}")
        return

    insert_query = """
        INSERT INTO artikel (nama_artikel, asal_artikel,
        link_artikel, tanggal_terbit, penulis_artikel)
        VALUES (%s, %s, %s, %s, %s)
    """
    execute_query(cursor, insert_query, articles,
    many=True)
    print(f"Inserted {len(articles)} articles from
    {issue_url}")
```

```

def delete_duplicates(cursor):
    delete_query = """
        DELETE a FROM artikel AS a
        INNER JOIN (
            SELECT link_artikel, MIN(id_artikel) AS min_id
            FROM artikel
            GROUP BY link_artikel
        ) AS b ON a.link_artikel = b.link_artikel AND
        a.id_artikel <> b.min_id
    """
    execute_query(cursor, delete_query)

def get_article_date(article_url):
    """Masuk ke halaman detail artikel dan ambil tanggal
    terbit."""
    try:
        response = requests.get(article_url)
        soup = BeautifulSoup(response.content,
                              "html.parser")

        published_element = soup.find("div", class_="item
published")
        if published_element:
            tanggal_text =
published_element.get_text(strip=True)
        else:
            published_element = soup.find("div",
class_="published")
            if published_element:
                tanggal_text =
published_element.get_text(strip=True)
            else:
                print(f"Tanggal terbit tidak ditemukan di
{article_url}")
                return None

        # Hapus semua huruf selain angka dan tanda minus
        (tanggal)
        tanggal_clean = re.sub(r"^[^d\~]", "",
tanggal_text)

        # Debug optional:
        # print(f"Tanggal dari {article_url} setelah
dibersihkan: {tanggal_clean}")

        return datetime.datetime.strptime(tanggal_clean,
"%Y-%m-%d").date()

    except Exception as e:
        print(f"Error getting date for {article url}:

```

```

        return None

def scrape_articles(issue_url, journal_id):
    try:
        page = requests.get(issue_url)
        soup = BeautifulSoup(page.content, "html.parser")
        articles = []

        # Mencari elemen yang berisi daftar artikel
        article_elements = soup.find_all("div",
class_="obj_article_summary")
        for article in article_elements:
            title_element = article.find("h3",
class_="title")
            link_element = title_element.find("a") if
title_element else None

            nama_artikel = (
                link_element.get_text(strip=True) if
link_element else "No Title"
            )
            link_artikel = link_element["href"] if
link_element else "No Link"
            penulis_element = article.find("div",
class_="authors")
            penulis_artikel = (
                penulis_element.get_text(strip=True)
                if penulis_element
                else "No Authors"
            )

            # Ambil tanggal dari halaman detail artikel
            tanggal_terbit =
get_article_date(link_artikel)

            articles.append(
                (
                    nama_artikel,
                    journal_id,
                    link_artikel,
                    tanggal_terbit,
                    penulis_artikel,
                )
            )

        return articles
    except Exception as e:
        print(f"Error scraping articles from {issue_url}:
{e}")
    return []

```

```

def scrape_journal(cursor, base_url, journal_id):
    try:
        page = requests.get(base_url)
        soup = BeautifulSoup(page.content, "html.parser")

        # Mencari elemen yang berisi daftar edisi (issue)
        issue_elements = soup.find_all("div",
class_="obj_issue_summary")
        for issue in issue_elements:
            issue_link_element = issue.find("a")
            if issue_link_element:
                issue_url = issue_link_element["href"]
                articles = scrape_articles(issue_url,
journal_id)
                insert_articles(cursor, articles,
issue_url)
            else:
                print(f"No link found for issue in
{base_url}")
        except Exception as e:
            print(f"Error scraping journal {base_url}: {e}")

def main():
    db = connect_to_database()
    cursor = db.cursor()

    # Hapus jurnal yang tidak diinginkan sebelum scraping
    cursor.execute(
        """
        DELETE FROM jurnal
        WHERE archive_jurnal IN

('https://journal.walisongo.ac.id/index.php/PJIS/issue/arc
hive',

'https://journal.walisongo.ac.id/index.php/rjtpd/issue/arc
hive')
        """
    )
    delete_duplicates(cursor)

    # Ambil daftar link archive jurnal
    cursor.execute("SELECT id_jurnal, archive_jurnal FROM
jurnal")
    journals = cursor.fetchall()

    for journal_id, base_url in journals:
        scrape_journal(cursor, base_url, journal_id)

    delete_duplicates(cursor)
    print("Duplicate rows deleted from artikel table.")

```

```
db.close()
print("Execution time:", time.time() - start_time)

if __name__ == "__main__":
    main()
```

