

TUGAS AKHIR

**MODEL ANALISIS APLIKASI SIIZIN MENGGUNAKAN
CLEAN CODE BERDASARKAN *SMELLS AND HEURISTICS***



Oleh:

Muhammad Irfan

2024240078

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER DAN REKAYASA
UNIVERSITAS MULTI DATA PALEMBANG
PALEMBANG
2024**

**Fakultas Ilmu Komputer dan Rekayasa
Universitas Multi Data Palembang**

Program Studi Sistem Informasi
Tugas Akhir Sarjana Komputer
Semester Gasal Tahun 2023/2024

**Model Analisis Aplikasi Siizin Menggunakan *Clean Code*
Berdasarkan *Smells and Heuristics***

Muhammad Irfan 2024240078

Abstrak

Penelitian ini membahas penerapan prinsip *Clean Code* pada pengembangan aplikasi SIIZIN berbasis CodeIgniter 3 dengan menerapkan analisis *smells* dan *heuristics*. Pada pengamatan awal, *source code* aplikasi SIIZIN menunjukkan beberapa masalah seperti format kode yang berantakan, penamaan variabel yang tidak sesuai standar dan tidak konsisten, serta komentar yang tidak diperlukan, yang semuanya menghambat pemahaman dan pemeliharaan kode. Dalam penelitian ini, ditemukan *smells code* pada kategori *Names*, *Comments*, dan *Generals*, seperti *N1: Choose Descriptive Names*, *C5: Commented-Out Code*, *C3: Redundant Comment*, *G5: Duplication*, *G20: Function Names Should Say What They Do*, dan *G12: Clutter*. Melalui analisis mendalam dan merancang kode sesuai dengan prinsip *clean code* yang berdampak meningkatkan keterbacaan dan efisiensi kode, penghapusan duplikasi, serta penjelasan fungsi sesuai dengan namanya. Penelitian ini juga menghilangkan komentar yang tidak berguna untuk meningkatkan kejelasan dan pemeliharaan kode. Hasilnya merancang rekomendasi *source code* yang telah memenuhi prinsip *clean code* menurut Robert C. Martin, atau Uncle Bob serta memudahkan pemeliharaan dan pengembangan aplikasi SIIZIN, memastikan keberlanjutan dan efektivitasnya dalam manajemen izin.

Kata kunci: *Clean Code*, *Smells and Heuristics*, Codeigniter, Universitas Multi Data Palembang, SIIZIN



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital yang semakin maju, perangkat lunak atau aplikasi telah menjadi hal yang penting dalam berbagai aspek kehidupan, mulai dari pemerintahan, pendidikan, hingga bisnis. Aplikasi perangkat lunak tidak hanya perlu berfungsi dengan baik, tetapi juga harus mudah dipahami, aman, dan mampu beradaptasi dengan perubahan. Prinsip *clean code* atau kode bersih menjadi landasan penting dalam menghadapi tuntutan ini. *Clean code* menekankan pembuatan kode yang tidak hanya berfungsi, tetapi juga mudah dibaca dan dimengerti oleh berbagai pengembang yang mungkin terlibat dalam proyek tersebut.

Penerapan *clean code* secara efektif dibutuhkan panduan yang jelas, *Smells* digunakan untuk menunjukkan tanda-tanda potensial dalam kode yang mungkin tidak bersih atau memerlukan perbaikan, sementara *heuristics* memberikan pedoman praktis untuk membantu pengembang membuat keputusan yang lebih baik dalam merancang dan merawat kode. Dengan mengidentifikasi *smells* dan mengikuti *heuristics*, pengembang dapat dengan lebih mudah mengenali masalah dalam kode mereka dan mengambil langkah-langkah yang tepat untuk meningkatkan kualitas dan keterbacaan kode. (Robert C. Martin, 2008).

Penelitian ini dilakukan di Universitas Multi Data Palembang atau UMDP. UMDP awalnya berdiri sebagai lembaga kursus komputer di kota Palembang. Seiring berjalannya waktu, UMDP telah berkembang menjadi sebuah universitas yang menawarkan berbagai program studi setara Diploma 1. Pada tanggal 9 April 2021, UMDP mengalami penggabungan yang signifikan, yang didasari oleh Surat Keputusan Menteri Pendidikan dan Kebudayaan Republik Indonesia No. 125/E/O/2021. Dalam penggabungan ini, lembaga-lembaga pendidikan seperti AMIK MDP, STMIK GI MDP, dan STIE MDP digabungkan menjadi satu entitas yang dikenal sebagai Universitas Multi Data Palembang.

Universitas Multi Data Palembang memiliki aplikasi dengan peran yang penting, yaitu Aplikasi SIIZIN, aplikasi SIIZIN menjadi alat yang berguna untuk manajemen izin bagi karyawan dan dosen di lingkungan universitas. Aplikasi ini menyediakan fitur-fitur yang mencakup empat kategori izin utama, yakni izin keluar, izin tidak masuk, izin telat, dan izin lainnya. Keempat kategori izin ini memungkinkan pengguna untuk mengatur dan melacak izin dengan lebih efisien dan terstruktur. Selain itu, Aplikasi SIIZIN juga memiliki kemampuan untuk membuat rekapitulasi pemberitahuan izin tidak masuk dan izin lainnya, menjadikannya alat yang sangat berguna dalam pengelolaan kehadiran dan izin di lingkungan Universitas Multi Data Palembang.

Aplikasi SIIZIN di UMDP menjadi aplikasi yang cocok untuk menerapkan prinsip-prinsip *clean code* karena saat aplikasi SIIZIN belum menerapkan prinsip *clean code* pada pengembangannya, lalu pada pengamatan awal pada *source code*

aplikasi SIIZIN ditemukan format kode yang berantakan, penamaan variabel yang tidak sesuai standar dan tidak konsisten, serta komentar yang tidak diperlukan, semuanya menghambat pemahaman dan pemeliharaan kode karena itulah aplikasi SIIZIN cocok untuk menerapkan prinsip *clean code*, dengan menerapkan *clean code* dapat meningkatkan kualitas, keamanan, dan pemeliharaan aplikasi tersebut, memudahkan pengembang lain dalam memahami dan mengubah kode, serta menjaga aplikasi tetap berjalan dengan baik dan aman seiring berjalannya waktu maka dari itu diangkat judul penelitian ini menjadi " Model Analisis Aplikasi Siizin Menggunakan Clean Code Berdasarkan Smells And Heuristics ".

1.2 Rumusan Masalah

1. Bagaimana cara mengidentifikasi kode yang buruk dengan memanfaatkan *Smells and Heuristics*?
2. Bagaimana menilai dan menerapkan prinsip-prinsip *clean code* dalam aplikasi SIIZIN untuk menentukan apakah kode tersebut sudah memenuhi standar kebersihan atau belum?
3. Bagaimana mengembangkan *prototype souce code* aplikasi SIIZIN menggunakan prinsip dasar dari *Clean Code*?

1.3 Ruang Lingkup

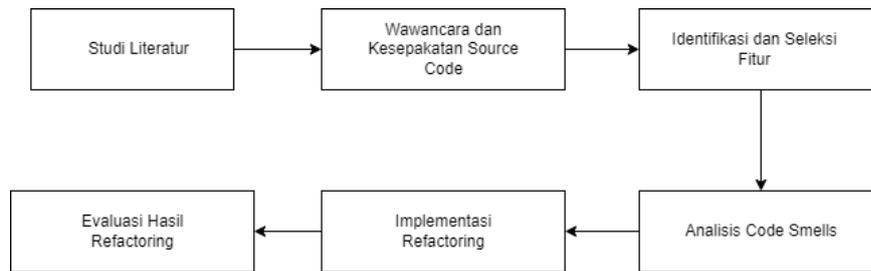
Dalam penelitian ini ditentukan ruang lingkup atau batasan masalah sebagai berikut.

1. Fokus penelitian ini terbatas pada evaluasi dan perbaikan *source code* tanpa mengganggu atau mengubah fitur-fitur yang sudah ada.

2. Aplikasi yang digunakan adalah SIIZIN di Universitas Multi Data Palembang.
3. Prinsip *Clean Code* yang diterapkan mengacu pada buku *Clean Code* karya Robert C. Martin.
4. Fitur yang dianalisis pada penelitian ini adalah izin keluar, izin lainnya, izin telat dan izin tidak masuk.
5. Analisis *code smells* hanya mencakup kategori *Comments, Functions, General, Names*.
6. Hasil penelitian ini hanya memberikan rekomendasi *prototype source code* sesuai prinsip *clean code* menurut Robert C. Martin.

1.4 Metodologi

Penelitian ini memberikan solusi melakukan *refactoring* pada aplikasi SIIZIN dengan metodologi yang digunakan terdiri dari beberapa tahap, yaitu tahap studi literatur, wawancara, kesepakatan mengenai source code, identifikasi dan seleksi fitur, analisis code smells, implementasi *refactoring*, dan evaluasi hasil *refactoring*. Tahap-tahap ini telah disesuaikan dengan kebutuhan penelitian berdasarkan buku "Clean Code: A Handbook of Agile Software Craftsmanship" yang dikembangkan oleh Robert C. Martin.



Gambar 1.1 Metodologi Penelitian

1. Studi Literatur

Tahap awal penelitian ini dimulai dengan melakukan studi literatur, dasar teori diperoleh dari sumber-sumber seperti jurnal internasional, buku, artikel, dan penelitian sebelumnya. Sumber-sumber ini digunakan sebagai panduan untuk penelitian ini.

2. Wawancara dan Kesepakatan Source Code

Tahap berikutnya adalah wawancara dan kesepakatan *source code* pada aplikasi yang akan diteliti. Ini melibatkan wawancara dengan pengembang aplikasi untuk memahami secara mendalam tentang aplikasi tersebut. Informasi ini akan menjadi landasan selama proses penelitian tentang *Clean Code* pada aplikasi yang bersangkutan.

3. Identifikasi dan Seleksi Fitur

Tahap ini, dilakukan identifikasi dan pemilihan fitur yang akan dilakukan perbaikan. Pemilihan ini dipertimbangkan berdasarkan pentingnya dan signifikannya fitur pada aplikasi SIIZIN.

4. Analisis Code Smells

Tahap ini melibatkan analisis mendalam terhadap kode sumber aplikasi untuk mengidentifikasi segala indikasi *Code Smells* yang mungkin ada. Analisis ini penting untuk menentukan area yang memerlukan perbaikan.

5. Implementasi Refactoring

Setelah mengidentifikasi area yang memerlukan perbaikan, langkah selanjutnya adalah menerapkan *refactoring*. Selama proses ini, prinsip-prinsip heuristik yang telah ditetapkan digunakan sebagai panduan. Setelah dilakukan implementasi, maka akan diperoleh rekomendasi source code aplikasi yang telah sesuai dengan prinsip *clean code*.

6. Evaluasi Hasil *Refactoring*

Tahap terakhir melibatkan evaluasi menyeluruh terhadap hasil *refactoring* yang telah dilakukan. Peningkatan kualitas kode dievaluasi dengan mengukur hasil dari perubahan yang telah diimplementasikan, dimana akan dilakuakn perbandingan sebelum dan sesudah diterapkan prinsip *clean code*.

1.5 Sistematika Penulisan

Penyusunan laporan skripsi ini dengan lebih terstruktur, penulisan dibagi menjadi lima bab utama. Setiap bab memiliki perannya sendiri dalam membantu pembaca memahami isi penelitian dengan jelas.

BAB I: PENDAHULUAN

Pada bab ini, akan diuraikan latar belakang masalah yang menjadi fokus penelitian, perumusan masalah yang akan diselesaikan, maksud dan tujuan penelitian, batasan masalah yang relevan, metodologi penelitian yang digunakan, serta sistematika penulisan yang akan memberikan panduan tentang isi dan urutan pembahasan dalam tugas akhir ini.

BAB II: LANDASAN TEORI

Bab ini berisikan teori-teori yang mendukung *clean code* dalam pengembangan perangkat lunak serta definisi prinsip *clean code* dari berbagai sumber yang berkaitan dengan penelitian ini. Tujuan bab ini adalah memberikan pemahaman yang kuat tentang landasan teoritis yang mendukung praktik *clean code* dalam pengembangan perangkat lunak.

BAB III: ANALISIS

Dalam bab ini, akan dibahas analisis *clean code* pada aplikasi SIIZIN menggunakan analisis *smells code* beserta alasan mengapa merefactor fitur yang dipilih.

BAB IV: IMPLEMENTASI

Bab ini akan mengulas implementasi dalam bahasa pemrograman, termasuk penerapan kebutuhan yang telah dianalisis. Selain itu, akan disajikan perbandingan antara kode yang telah *refactor* dan yang belum *refactor*, disertai dengan penjelasannya.

BAB V: KESIMPULAN DAN SARAN

Pada bab terakhir ini, akan diuraikan kesimpulan yang diperoleh dari hasil penelitian tugas akhir. Selain itu, saran-saran untuk pengembangan aplikasi

di masa yang akan datang akan diberikan, memberikan pandangan yang berguna untuk perbaikan dan pengembangan lebih lanjut.

1.6 Tujuan dan Manfaat

Penelitian ini memiliki dua tujuan utama, yaitu mengidentifikasi code smells dalam aplikasi SIIZIN dan menerapkan prinsip *clean code* untuk perbaikan. Identifikasi code smells akan membantu memahami dampak terhadap kualitas kode, sementara penerapan *clean code* bertujuan membuat *source code* yang lebih bersih dan mudah dipelihara.

Penelitian ini memberikan manfaat penting melalui dua tujuan utamanya. Pertama, dengan mengidentifikasi dan menganalisis code smells dalam aplikasi SIIZIN akan memberikan manfaat untuk Universitas Multi Data Palembang dengan meningkatkan kualitas kode aplikasi dan meningkatkan efisiensi pemeliharaan kode. Kedua, dengan menerapkan prinsip *clean code*, penelitian ini memberikan perbaikan kode aplikasi SIIZIN, menghasilkan *source code* yang lebih bersih dan mudah dipelihara. Selain manfaat teknisnya, penelitian ini berpotensi memberikan kontribusi pada pengembangan profesional pengembang dan praktik-praktik terbaik dalam industri perangkat lunak, serta memberikan rekomendasi *source code* yang bersih untuk SIIZIN.



DAFTAR PUSTAKA

- Andi Jamiati Paramita, Andi Hutami Endang, D.A.K. (2022) “PENDETEKSIAN CODE SMELL PADA WEBSITE PERUSAHAAN,” *Jurnal Ilmiah Indonesia*, 7(1).
- CodeIgniter Foundation (2023) “CodeIgniter Documentation.” Tersedia pada: <https://www.codeigniter.com/userguide3/general/welcome.html>.
- Firdaus, M.F., Priyambadha, B. dan Pradana, F. (2018) “Pembangunan Sistem Untuk Pendeteksian Code Smells Refused Bequest,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(12), hal. 6722–6728. Tersedia pada: <http://j-ptiik.ub.ac.id>.
- HADIANSYAH, F. (2020) “Optimasi Maintainability Menggunakan Metode Clean Code Pada Sistem Informasi Museum Mandhilaras,” hal. 1–6.
- Inayah, F.I. dan Idris, M. (2021) “Implementasi Clean Code pada Pengembangan Berbasis Web,” *Automata*, 2(2), hal. 113–116.
- Lacerda, G. *dkk.* (2020) “Code smells and refactoring: A tertiary systematic review of challenges and observations,” *Journal of Systems and Software*, 167. Tersedia pada: <https://doi.org/10.1016/j.jss.2020.110610>.
- Latte, B., Henning, S. dan Wojcieszak, M. (2019) “Clean code: on the use of practices and tools to produce maintainable code for long-living software,” *CEUR Workshop Proceedings*, 2308, hal. 96–99.
- Ljung, K. (2021) “Clean Code in Practice Developers’ perception of clean code,” *Diva* [Preprint]. Tersedia pada: www.bth.se.
- Ljung, K. dan Gonzalez-Huerta, J. (2022) “‘To Clean Code or Not to Clean Code’ A Survey Among Practitioners,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13709 LNCS, hal. 298–315. Tersedia pada: https://doi.org/10.1007/978-3-031-21388-5_21.
- Microsoft (2023) “Visual Studio Code Documentation.” Tersedia pada: <https://code.visualstudio.com/docs>.
- Randi V. Palit., Yaulie D.Y. Rindengan, ST., MM., MSc., Arie S.M. Lumenta, ST.,

- M. (2015) “Rancangan Sistem Informasi Keuangan Gereja Berbasis Web Di Jemaat GMIM Bukit Moria Malalayang,” *E-Journal Teknik Elektro dan Komputer*, 4(ISSN: 2301-8402 1), hal. 98–103. Tersedia pada: <https://doi.org/10.33084/jsakti.v3i1.1770>.
- RIZALDHI, D.A. (2019) “Implementasi Clean Code Dan Design Pattern Untuk Meningkatkan Maintainability Pada Aplikasi Content Marketing,” hal. 3–13. Tersedia pada: <http://repository.usu.ac.id/bitstream/123456789/23790/4/Chapter I.pdf>.
- Robert C. Martin (2008) *Clean Code A Handbook of Agile Software Craftmanship, Learn CakePHP*. Tersedia pada: https://doi.org/10.1007/978-1-4842-1212-7_3.
- Windler, C. dan Daubois, A. (2022) *Clean Code in PHP: Expert tips and best practices to write beautiful, human-friendly, and maintainable PHP*.
- Yana, D. *dkk.* (2023) “Clean Code In Practice: Challenges and Opportunities,” *SSRN* [Preprint].

