

## **TUGAS AKHIR**

**MODEL ANALISIS APLIKASI SIMUDAH MENGGUNAKAN *CLEAN CODE* BERDASARKAN *CODE SMELLS* DAN *HEURITICS***



**Oleh:**

**Marcelo Amazona**

**2024240036**

**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 SIMUDAH MENGGUNAKAN *CLEAN CODE* BERDASARKAN *CODE SMELLS* DAN *HEURISTICS***

Marcelo Amazona    2024240036

**Abstrak**

Pengembangan perangkat lunak dalam era kemajuan teknologi menuntut penerapan prinsip *Clean Code* untuk memastikan keterbacaan dan efisiensi kode. Penelitian ini fokus pada aplikasi SIMUDAH dari Universitas Multi Data Palembang, dikembangkan dengan *framework Laravel*. Meskipun *framework* tersebut memberikan struktur yang terorganisir, SIMUDAH menghadapi kendala pemeliharaan, seperti singkatan *variabel* dan *dead code*. Penelitian ini bertujuan menerapkan *Clean Code* dengan mengidentifikasi dan memperbaiki *smells code*, menggunakan *heuristics* seperti *Single Responsibility Principle* dan *Encapsulation*. Diharapkan, penelitian ini akan meningkatkan kualitas dan pemeliharaan SIMUDAH, menjadikannya solusi yang relevan dan efisien di lingkungan kegiatan mahasiswa.

**Kata kunci:** *Clean Code*, *Code Smells*, *Laravel*, Pemeliharaan Kode, Efisiensi Kode, Universitas Multi Data Palembang, SIMUDAH.



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Di era yang semakin maju ini, pengembangan perangkat lunak menjadi semakin kritis dan penting, memerlukan fondasi yang kuat dalam penerapan prinsip-prinsip Clean Code. Konsep Clean Code, sebagaimana didefinisikan dalam buku "Clean Code" karya Robert C. Martin, menjadi landasan utama dalam membangun perangkat lunak yang tidak hanya inovatif tetapi juga berkualitas tinggi. Dalam menghadapi dinamika zaman yang terus berkembang, prinsip-prinsip ini menjadi kunci untuk menjaga keterbacaan, pemeliharaan, dan efisiensi dari suatu kode (Putro dan Liem, 2010).

Dalam konteks zaman yang semakin maju, "Smells code" atau bau kode menjadi petunjuk awal bahwa suatu kode mungkin tidak memenuhi standar Clean Code. Fenomena ini bisa mencakup fungsi yang terlalu panjang, penggunaan variabel dengan nama yang tidak jelas, komentar yang kurang informatif, atau kelas yang terlalu kompleks. Seiring dengan kemajuan teknologi, pentingnya menerapkan prinsip-prinsip Clean Code semakin menjadi fokus utama dalam menjaga relevansi dan kualitas perangkat lunak (Putro dan Liem, 2010).

Namun, tantangan muncul ketika penulis menerapkan prinsip-prinsip ini

dalam konteks yang terus berkembang pesat. Heuristics atau aturan praktis, seperti Single Responsibility Principle dan Encapsulation, menjadi panduan krusial dalam menghadapi kompleksitas pengembangan perangkat lunak di era yang semakin canggih ini (Abdul dan Lidinillah, 2008). Heuristics ini tidak hanya membantu pengembang untuk membuat keputusan yang lebih baik dalam menulis, membaca, dan memelihara kode, tetapi juga menjadi kunci sukses dalam menghadapi tuntutan pengembangan perangkat lunak di zaman modern ini. Dalam menyikapi tuntutan ini, Universitas Multi Data Palembang (UMDP) muncul sebagai salah satu lembaga pendidikan yang progresif. Sebagai yayasan pendidikan, UMDP memiliki fokus yang kuat pada transfer pengetahuan, keterampilan, dan nilai dari pengajar kepada pelajar. Dalam upaya untuk memudahkan kegiatan mahasiswa, UMDP merancang dan mengembangkan aplikasi bernama SIMUDAH (Sistem Informasi MDP Unit Kegiatan Mahasiswa dan Himpunan Mahasiswa Jurusan). SIMUDAH memiliki peran yang sangat penting dalam mengelola berbagai kegiatan mahasiswa, termasuk absensi dan pelaporan.

Alasan penggunaan aplikasi SIMUDAH menjadi semakin relevan karena aplikasi ini dikembangkan menggunakan framework Laravel. Framework Laravel merupakan salah satu framework PHP yang populer, terkenal akan kejelasan dan kelincuhan kodenya. Dengan menggunakan Laravel, SIMUDAH tidak hanya menjadi solusi yang efisien dalam mengelola kegiatan mahasiswa, tetapi juga memastikan bahwa basis kode aplikasi ini mematuhi prinsip-prinsip Clean Code. Penggunaan framework Laravel memberikan struktur yang

terorganisir dan standar pengkodean yang jelas, sehingga memudahkan dalam memahami, mengelola, dan memelihara source code SIMUDAH.

Namun, seiring dengan manfaat yang ditawarkan, SIMUDAH menghadapi beberapa permasalahan, terutama dalam hal pemeliharaan. Kode yang penuh dengan singkatan variabel dan dead code menjadi hambatan utama dalam pemahaman dan pemeliharaan sistem informasi ini. Keadaan ini menciptakan dampak negatif terhadap keterbacaan dan pemeliharaan SIMUDAH, mengingat semakin kompleksnya kebutuhan dan ekspektasi dalam pengembangan perangkat lunak di era modern.

Dengan menyadari tantangan ini, penulis mengusulkan penerapan Clean Code pada SIMUDAH. Fokusnya adalah mengidentifikasi dan memperbaiki smells code, serta menerapkan heuristics untuk menciptakan kode yang lebih bersih, mudah dipahami, dan mudah dikelola. Melalui pendekatan ini, diharapkan SIMUDAH dapat terus menjadi solusi yang relevan, efisien, dan andal dalam mengelola kegiatan mahasiswa di tengah kemajuan zaman yang semakin pesat. Dengan demikian, judul penelitian ini menjadi "Model analisis aplikasi simudah menggunakan *clean code* berdasarkan *code smells* dan *heuritics* "

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, masalah yang ditemukan berikut.

1. Apakah aplikasi simudah sudah memiliki standarisasi *Cleancode*?
2. Bagaimana menerapkan prinsip *Clean Code* dalam aplikasi simudah terutama dalam hal kualitas, pemeliharaan, dan efisiensi kode?

### 1.3 Ruang Lingkup

Dalam penelitian ini ditentukan ruang lingkup atau batasan masalah sebagaiberikut.

1. Fokus pada penelitian ini yaitu pada *source code* saja, tanpa mengubah fiturfitur yang ada pada Aplikasi SIMUDAH.
2. Aplikasi yang digunakan aplikasi Simudah di Universitas Multi Data Palembang.
3. Metode *Clean Code* yang digunakan buku clean code Robert C Martin
4. Fokus *Refactoring* pada Fitur *Absensi*, Fitur Jadwal, Fitur Anggota, Fitur Laporan, Fitur Laporan Mahasiswa dan Login.
5. Analisis mencakup kategori – kategori berikut: Komentar, Fungsi, umum dan nama.
6. Hasil penelitian ini hanya memberikan rekomendasi protipe kode sumber sesuai standar clean code menurut Robert C. Martin.

### 1.4 Tujuan dan Manfaat

#### 1.4.1 Tujuan

Penelitian ini bertujuan untuk menerapkan model analisis aplikasi berfokus pada prinsip-prinsip Clean Code dengan menggunakan pendekatan Code Smells dan Heuristics. Fokus utama penelitian adalah pada aplikasi SIMUDAH di Universitas Multi Data Palembang, dengan upaya meningkatkan kualitas, keterbacaan, dan pemeliharaan kode

### **1.4.2 Manfaat**

Penelitian ini diharapkan memberikan manfaat besar pada aplikasi SIMUDAH di Universitas Multi Data Palembang. Implementasi prinsip Clean Code dan pendekatan Code Smells serta Heuristics diharapkan meningkatkan kualitas, keterbacaan, dan pemeliharaan kode. Dengan kode yang bersih, mempermudah pengembang baru dan tim pengembangan, sambil meningkatkan efisiensi pemeliharaan sistem. Manfaat lainnya melibatkan peningkatan produktivitas pengembang, mempercepat pengembangan, dan mengurangi waktu pemeliharaan. Akhirnya, penelitian ini bertujuan memberikan pengalaman pengguna yang lebih baik melalui pengembangan fitur baru dan perbaikan yang lebih cepat pada SIMUDAH.

### **1.5 Metodologi Penelitian**

Dalam Proposal Penelitian Tugas Akhir ini, jenis penelitian yang digunakan adalah pendekatan kualitatif. Terdapat sejumlah tahap penelitian, masing-masing memiliki kegiatan yang didukung oleh penggunaan metode tertentu. Metode penelitian yang diterapkan dalam penyusunan laporan Proposal Tugas Akhir ini didasarkan pada buku “Clean Code: A Handbook of Agile Software Craftsmanship” yang dikembangkan oleh Robert C. Martin, dan kemudian disesuaikan dengan kebutuhan penelitian.

### 1. Studi Literatur

Dasar teori didapatkan peneliti melalui jurnal-jurnal internasional, buku, artikel serta beberapa penelitian sebelumnya yang dapat digunakan untuk panduan terhadap penelitian ini.

### 2. Wawancara dan kesepakatan *source code*

Tahap ini melakukan wawancara terhadap Pengembang aplikasinya untuk membantu selama penelitian Clean Code pada Aplikasi yang telah disepakati.

### 3. Identifikasi dan seleksi *fitur refactoring*

Dalam Tahap ini, Kami melakukan pemilihan fitur yang paling sesuai dengan penelitian ini, dengan mempertimbangkan perbaikansignifikan dalam kode.

### 4. Analisis *Code Smells*

Kami melakukan analisis mendalam terhadap kode sumber untuk mengidentifikasi segala indikasi *Code Smells* yang mungkin ada. Ini adalah Langkah penting untuk menentukan area yang memerlukan perbaikan.

## 5. Implementasi *Refactoring* dengan Prinsip *Heuristik*

Kami menerapkan *refactoring* dengan berpegang teguh pada prinsip prinsip *heuristic* yang telah kami tetapkan. Ini membantu memastikan bahwa perubahan yang kami buat tidak hanya meningkatkan kode tapi mempertahankan integritasnya.

## 6. Evaluasi Hasil *Refactoring*

Akhirnya, kami melakukan evaluasi menyeluruh terhadap hasil *refactoring*, dengan mengukur peningkatan kualitas kode.



**Gambar 1.1 Metodologi Penelitian**

## 1.6 Sistematika Penulisan

Untuk memudahkan pembahasan dalam laporan skripsi ini, sistematika penulisan dibagi menjadi 5 (lima) bab yang terdiri dari:

### **BAB 1 PENDAHULUAN**

Pada bab ini, akan dibahas latar belakang masalah yang diambil, perumusan masalah, maksud dan tujuan, batasan masalah, metodologi penelitian, dan sistematika penulisan.

### **BAB 2 LANDASAN TEORI**

Pada bab ini, akan dibahas landasan teori yang digunakan dalam mendukung

Analisis clean code termasuk teori-teori yang berkaitan dengan teknologi dan metode yang digunakan dalam penelitian ini.

### **BAB 3 ANALISIS DAN PEMETAAN CLEAN CODE PADA APLIKASI**

Pada bab ini, akan didiskusikan analisis kebutuhan dalam konteks Clean Code pada aplikasi ini. Selain itu, akan dijelaskan faktor-faktor yang memengaruhi kualitas keseluruhan aplikasi. Bagian-bagian file kode yang akan menjalani proses refactoring juga akan diuraikan, bersama dengan tahapan yang perlu ditempuh dalam melakukan refactoring tersebut.

### **BAB 4 IMPLEMENTASI**

Pada bab ini, akan dibahas implementasi dalam bahasa pemrograman, termasuk implementasi kebutuhan dalam analisis. Selain itu, akan dijelaskan perbandingan antara code yang telah diperbaiki dan belum diperbaiki beserta penjelasannya.

### **BAB 5 KESIMPULAN DAN SARAN**

Pada bab ini, akan dibahas kesimpulan yang diperoleh dari hasil penulisan tugas akhir. Selain itu, akan diberikan saran mengenai pengembangan aplikasi untuk masa yang akan datang



## DAFTAR PUSTAKA

- Abdul, D. dan Lidinillah, M. (2008) “Heuristik dalam Pemecahan Masalah Matematika dan Pembelajarannya di SD,” hal. 1–11.
- C.Martin, R. (2016) *Clean Code, Learn CakePHP*. Tersedia pada: [https://doi.org/10.1007/978-1-4842-1212-7\\_3](https://doi.org/10.1007/978-1-4842-1212-7_3).
- Djoni Haryadi Setiabudi dan Ibnu Gunawan (2003) “Studi Penggunaan Visual Studio 6.0 Untuk Pengembangan Sistem Informasi Berkelas Enterprise,” *Jurnal Informatika*, 4(1), hal. 27–34. Tersedia pada: <http://puslit2.petra.ac.id/ejournal/index.php/inf/article/view/15829>.
- Hadiansyah, F. (2019) “Pendahuluan 1.1,” *Skripsi*, hal. 1–6.
- Inayah, F.I. dan Idris, M. (2021) “Implementasi Clean Code pada Pengembangan Berbasis Web,” *Automata*, 2(2), hal. 113–116.
- Kazi, L., Mihajlović, S. dan Bhatt, M. (2021) “Clean Code Quality Attributes and Measurements: an Initial Review,” (October 2021). Tersedia pada: <https://www.researchgate.net/publication/366578085>.
- Lacerda, G. *dkk.* (2020) “Code smells and refactoring: A tertiary systematic review of challenges and observations,” *Journal of Systems and Software*, 167(April). Tersedia pada: <https://doi.org/10.1016/j.jss.2020.110610>.
- Ljung, K. (2021) “Clean Code in Practice,” *Blekinge Institute of Technology* [Preprint]. Tersedia pada: [www.bth.se](http://www.bth.se).
- Pamungkas, Y.B. (2019) “Pemrograman Programmable Logic Controller ( Plc ) Berbasis ‘ Clean Code ’ Skripsi.”
- Paramita, A.J., Endang, A.H. dan Khairunnisa, D.A. (2022) “Pendeteksian Code Smell pada Website Perusahaan,” *Syntax Literate ; Jurnal Ilmiah Indonesia*, 7(1), hal. 53. Tersedia pada: <https://doi.org/10.36418/syntax-literate.v7i1.6019>.
- Parera, S., Sukmana, H.T. dan Wardhani, L.K. (2016) “Application of genetic algorithm for class scheduling (case study: faculty of science and technology UIN Jakarta),” *2016 4th International Conference on Cyber and IT Service Management*, hal. 1–5. Tersedia pada: <https://doi.org/10.1109/CITSM.2016.7577525>.

- Petra, U.K. (2016) “Tingkat hutang (DEB) ,” hal. 26–32.
- Putro, H.P. dan Liem, I. (2010) “Deteksi Code Smell pada Kode Program dalam Representasi AST dengan Pendekatan By Rules,” *Seminar Nasional Aplikasi dan Teknologi Informasi*, 2010(Snati), hal. 6.
- Rahmawati, A.F. dan Susetyo, Y.A. (2023) “Analisis Quality Code Menggunakan Sonarqube Dalam Suatu Aplikasi Berbasis Laravel,” *IT-Explore: Jurnal Penerapan Teknologi Informasi dan Komunikasi*, 2(2), hal. 99–103. Tersedia pada: <https://doi.org/10.24246/itexplore.v2i2.2023.pp99-103>.
- Rani, A., Kaur, H. dan Prof, A. (2014) “Detection of Bad Smells in Source Code According To Their Object Oriented Metrics,” *International Journal For Technological Research In Engineering ISSN*, 1(10), hal. 2347–4718.
- Virdus, V., Priyambadha, B. dan Soebroto, A.A. (2019) “Pembangunan Sistem Aplikasi Deteksi Code Smell berdasarkan Metrik Feature Envy,” 3(8), hal. 7500– 7506. Tersedia pada: <http://j-ptiik.ub.ac.id>.
- Yuliano, T. (2017) “Pengenalan PHP,” *Ilmu Komputer*, hal. 1–9.

