

# Internalisasi Prinsip Clean Code dan Validasi Logika pada Pengembangan Aplikasi Python bagi Mahasiswa Teknik Komputer

Afifah Farhanah Akadji<sup>1\*</sup>, Maharani Eka<sup>2</sup>

<sup>1\*</sup> Teknik Komputer, Universitas Negeri Gorontalo, Jl. Jend.  
Sudirman No.6, Kota Gorontalo, Indonesia

<sup>2</sup>Psikologi, Universitas Negeri Gorontalo, Jl. Jend.  
Sudirman No.6, Kota Gorontalo, Indonesia

\*Corresponding author. E-mail: [farhanah.akadji@ung.ac.id](mailto:farhanah.akadji@ung.ac.id);

## Abstrak

Menulis kode yang fungsional merupakan kompetensi dasar, namun menjamin kualitas internal kode (Clean Code) dan ketahanan program terhadap kesalahan pengguna adalah standar profesional yang harus dikuasai oleh mahasiswa Teknik Komputer. Kegiatan pengabdian ini bertujuan untuk menginternalisasi prinsip Clean Code dan validasi logika pada mahasiswa melalui metode Problem-Based Learning (PBL) dalam mata kuliah Dasar Pemrograman. Metode pelaksanaan meliputi orientasi standar koding, konstruksi aplikasi mandiri berbasis Python, serta sesi code review dan demonstrasi program. Sebanyak 12 prototype aplikasi dihasilkan, mencakup sistem manajemen data hingga simulasi transaksi digital. Hasil pengabdian menunjukkan bahwa mahasiswa berhasil mengimplementasikan aspek program persistence melalui struktur perulangan yang efisien serta membangun mekanisme validasi input untuk menjaga integritas data. Selain itu, kesadaran terhadap estetika antarmuka berbasis teks (UI/UX) terlihat dari penggunaan partisi menu yang sistematis. Pengabdian ini menyimpulkan bahwa pendekatan PBL efektif dalam menanamkan disiplin teknis dan profesionalisme pemrograman, yang menjadi fondasi penting bagi mahasiswa dalam menghadapi standar industri pengembangan perangkat lunak di masa depan.

**Kata kunci:** Clean Code, Dasar Pemrograman, Error Handling, Teknik Komputer, Validasi Logika

## Abstract

*Writing functional code is a fundamental competency, yet ensuring internal code quality (Clean Code) and program robustness against user errors are essential professional standards for Computer Engineering students. This community service initiative aims to internalize Clean Code principles and logic validation through the Problem-Based Learning (PBL) method in a Basic Programming course. The execution method involved coding standard orientation, independent Python-based application construction, and code review sessions. Twelve prototype applications were successfully developed, ranging from data management systems to digital transaction simulations. The results indicate that students were able to implement program persistence through efficient looping structures and build input validation mechanisms to maintain data integrity. Awareness of text-based interface aesthetics (UI/UX) was also evident through the use of systematic menu partitioning. This initiative concludes that the PBL approach effectively instills technical discipline and programming professionalism, providing a vital foundation for students to meet industry software development standards in the future.*

**Keywords:** Clean Code, Basic Programming, Error Handling, Computer Engineering, Logic Validation

Submitted: December 2025; Reviewed: December 2025; Accepted: December 2025

## 1. Pendahuluan

Dalam kurikulum Teknik Komputer, kemahiran dalam pemrograman merupakan kompetensi inti yang menentukan kesiapan mahasiswa dalam menghadapi industri teknologi informasi. Namun, penguasaan pemrograman tidak hanya terbatas pada kemampuan menyusun baris kode agar aplikasi dapat berjalan (functional), tetapi juga mencakup kualitas internal kode tersebut atau yang dikenal dengan prinsip Clean Code (Hermawan et al., 2021). Kode yang berkualitas harus memiliki keterbacaan yang baik, struktur logika yang efisien, dan mekanisme penanganan kesalahan (error handling) yang kuat untuk menjamin reliabilitas perangkat lunak.

Pada praktiknya, mahasiswa tingkat awal seringkali terjebak dalam pola pikir "pemrograman pragmatis", di mana fokus utama hanya tertuju pada luaran program tanpa memperhatikan kaidah penulisan kode yang standar (Kurniawan et al., 2022). Permasalahan umum yang ditemukan adalah kurangnya validasi terhadap input pengguna serta alur program yang tidak sistematis, yang berakibat pada kerentanan sistem saat dioperasikan (Rahman et al., 2024). Di Program Studi Teknik Komputer, Universitas Negeri Gorontalo, fenomena ini menjadi tantangan dalam membentuk karakter pengembang perangkat lunak yang disiplin dan profesional.

Untuk mengatasi tantangan tersebut, diterapkan pendekatan Problem-Based Learning (PBL) pada mata kuliah Dasar Pemrograman (Febriani et al., 2023). Melalui metode ini, mahasiswa tidak hanya diberikan teori, tetapi langsung dihadapkan pada tantangan untuk membangun prototype aplikasi mandiri menggunakan bahasa Python. Pendekatan ini mewajibkan mahasiswa untuk memenuhi kriteria teknis yang ketat, antara lain penerapan struktur pemilihan dan perulangan yang optimal, pembuatan antarmuka berbasis teks yang interaktif, serta kewajiban membangun sistem validasi input yang mampu menjaga integritas alur program (Tambunan et al., 2022).

Kegiatan pengabdian ini bertujuan untuk menginternalisasi prinsip-prinsip pengembangan perangkat lunak yang berkualitas melalui penugasan proyek yang terukur. Fokus utama pengabdian ditekankan pada bagaimana mahasiswa mampu mentransformasi logika berpikir mereka ke dalam kode yang rapi, tervalidasi, dan memiliki alur yang jelas. Melalui kegiatan ini, diharapkan mahasiswa memiliki fondasi yang kuat dalam standar penulisan program, sehingga mampu menghasilkan karya digital yang tidak hanya solutif tetapi juga memiliki kualitas teknis yang sesuai dengan standar profesional di bidang Teknik Komputer.

## 2. Metode

Kegiatan pengabdian ini dilaksanakan di Program Studi Teknik Komputer, Universitas Negeri Gorontalo dengan melibatkan 12 mahasiswa sebagai mitra sasaran. Metode yang digunakan adalah Problem-Based Learning (PBL) yang dirancang untuk mendorong disiplin penulisan kode melalui tahapan sebagai berikut (Sitanggang et al., 2023):

1. **Tahap Orientasi Standar Koding:** Dosen memberikan penguatan mengenai pentingnya Clean Code dalam bahasa Python, mencakup penamaan variabel yang deskriptif dan pentingnya struktur program yang modular menggunakan fungsi atau percabangan yang rapi (Aitbayeva et al., 2023).
2. **Tahap Eksplorasi Masalah:** Mahasiswa mengidentifikasi masalah harian yang unik untuk dijadikan proyek aplikasi. Dalam tahap ini, dosen memastikan setiap ide proyek memiliki kompleksitas yang cukup untuk menerapkan logika pemilihan dan perulangan (Yuliana et al., 2024).
3. **Tahap Konstruksi dan Validasi (Coding):** Mahasiswa membangun aplikasi dengan

kewajiban menerapkan sistem validasi input. Dosen memberikan pendampingan teknis melalui sesi konsultasi untuk memastikan program tidak hanya berjalan, tetapi juga tahan terhadap kesalahan masukan data oleh pengguna (error handling) (Indriasari et al., 2020).

4. **Tahap Review dan Demo:** Mahasiswa mendemonstrasikan hasil aplikasi di depan kelas untuk diuji ketahanan logikanya. Evaluasi dilakukan berdasarkan keterbacaan kode dalam laporan PDF dan kelancaran alur interaksi pengguna (UI berbasis teks) yang dihasilkan (Rokhim et al., 2023).

**Gambar 1**

*Dokumentasi kegiatan Review dan Demo*



Seluruh proses pelaksanaan pengabdian ini didokumentasikan melalui laporan teknis dalam format digital (PDF) yang disusun secara mandiri oleh setiap mahasiswa. Laporan tersebut memuat dokumentasi kode program (source code), penjelasan alur logika dalam bentuk flowchart, serta hasil tangkapan layar (screenshot) saat aplikasi dijalankan dengan berbagai skenario input. Melalui dokumentasi yang komprehensif ini, efektivitas penerapan metode PBL dalam meningkatkan kualitas penulisan kode dan validasi logika mahasiswa dapat dianalisis secara objektif pada bagian hasil.

### 3. Hasil

Kegiatan pengabdian pada Program Studi Teknik Komputer ini menghasilkan 12 produk perangkat lunak berbasis teks yang mencakup berbagai fungsi praktis. Berdasarkan evaluasi terhadap laporan PDF dan sesi demonstrasi program, seluruh mahasiswa (100%) berhasil mengimplementasikan aplikasi yang fungsional tanpa mengalami runtime error saat pengujian berlangsung.

Capaian teknis dari 12 proyek mahasiswa tersebut dirangkum dalam tabel evaluasi berikut:

**Tabel 1**

*Analisis Kualitas Internal Kode dan Validasi Logika Mahasiswa*

Nama Proyek Aplikasi	Fitur Unggulan (UI/UX)	Mekanisme Validasi Input	Status Keberhasilan
Kalender Interaktif	Partisi header sistematis	Validasi format bulan/tahun	Berhasil
Game Batu Gunting Kertas	Respon dialog interaktif	Validasi pilihan menu	Berhasil
Perpustakaan Sederhana	Layout daftar buku rapi	Cek ketersediaan stok buku	Berhasil
Aplikasi Kasir	Desain layout struk belanja	Validasi tipe data harga/jumlah	Berhasil
Game Tebak Angka	Petunjuk bantuan (clue)	Batasan rentang angka	Berhasil

SI Data Mahasiswa	Struktur menu CRUD	Validasi kelengkapan data	Berhasil
Mini ATM	Tampilan saldo terstandar	Validasi PIN & kecukupan saldo	Berhasil
Sistem Login	Penyamaran input (simulasi)	Validasi kesesuaian akun	Berhasil
Kalkulator Sederhana	Panel operasi aritmatika	Penanganan pembagian nol	Berhasil
Cek Plat Kendaraan	Indikator kode plat	Validasi format kode plat	Berhasil
Manajemen Nilai	Tabel rekapitulasi nilai	Validasi rentang nilai (0-100)	Berhasil
SI Data KTP	Format pengisian data rapi	Validasi jumlah karakter NIK	Berhasil

Berdasarkan Tabel 1, ditemukan bahwa mahasiswa Teknik Komputer memiliki kesadaran tinggi terhadap estetika antarmuka meskipun masih berbasis teks. Penggunaan partisi garis (----) secara konsisten diterapkan untuk memisahkan bagian header, isi, dan footer guna memudahkan pengguna memahami hierarki informasi, beberapa hasil program mahasiswa dapat dilihat pada Gambar 2 dan 3.

Dari sisi teknis, sebanyak 80% mahasiswa telah mengimplementasikan struktur perulangan tak terbatas (`while True`). Hal ini memungkinkan program untuk terus beroperasi dan melayani permintaan pengguna secara repetitif tanpa harus menjalankan ulang skrip dari awal, yang merupakan peningkatan signifikan dalam kenyamanan penggunaan aplikasi. Selain itu, aspek Validasi Logika telah diterapkan dengan baik melalui penggunaan fungsi pengecekan tipe data. Sebagai contoh, pada program yang melibatkan perhitungan saldo atau angka, mahasiswa telah menyisipkan logika untuk menolak input bertipe string dan meminta pengguna melakukan input ulang. Meskipun aplikasi yang dihasilkan belum sepenuhnya mengakomodasi kompleksitas variabel di dunia nyata secara menyeluruh, namun secara prinsip teknis, mahasiswa telah berhasil membangun fondasi kode yang kuat, bersih, dan tervalidasi.

## Gambar 2

*Antarmuka Program Cek Plat Kendaraan dengan Validasi Input*

```

SELAMAT DATANG DI PROGRAM CEK
PLAT KENDARAAN
-----

Menu:
1. Cek asal wilayah dari kode plat
2. Informasi program
3. Keluar
Pilih menu (1/2/3): 2

Program ini digunakan untuk mengecek asal wilayah kendaraan berdasarkan kode plat Indonesia.
CATATAN: Daftar kode kode plat dalam program ini TIDAK mencakup seluruh kode plat di Indonesia, hanya sebagian.

Menu:
1. Cek asal wilayah dari kode plat
2. Informasi program
3. Keluar
Pilih menu (1/2/3): 1

Masukkan kode plat (ketik 'kembali' untuk ke menu): dm
Asal wilayah: Gorontalo

Masukkan kode plat (ketik 'kembali' untuk ke menu): v
Asal wilayah: Kode plat tidak dikenali

Masukkan kode plat (ketik 'kembali' untuk ke menu): dk
Asal wilayah: Bali

Masukkan kode plat (ketik 'kembali' untuk ke menu): kembali

Menu:
1. Cek asal wilayah dari kode plat
2. Informasi program
3. Keluar
Pilih menu (1/2/3): 3
Terima kasih telah menggunakan program ini!

```

**Gambar 3***Tampilan menu dan struk dari Aplikasi Kasir*

```

=== PROGRAM KASIR SEDERHANA ===
1. Menghitung belanjaan
2. Keluar program
Pilih menu (1/2): 1
Berapa item yang dibeli? 2

=== Item 1 ===
Nama barang: kecap
Harga barang (Rp): 10000
Jumlah barang: 2

=== Item 2 ===
Nama barang: gula
Harga barang (Rp): 20000
Jumlah barang: 1

=== STRUK PEMBAYARAN ===
1. kecap x2 - Rp 20000
2. gula x1 - Rp 20000
-----
Subtotal : Rp 40000
Diskon   : Rp 0
Total    : Rp 40000

Ingin melakukan pembelian lagi? (Y/N): n
Terima kasih!

```

**4. Pembahasan**

Hasil pengabdian menunjukkan bahwa mahasiswa Teknik Komputer tidak hanya mampu memahami sintaks pemrograman Python, tetapi juga mulai menginternalisasi nilai-nilai kualitas dalam pengembangan perangkat lunak. Terdapat tiga poin utama yang menjadi fokus pembahasan dalam efektivitas metode PBL ini:

1. Estetika UI/UX sebagai Bentuk Komunikasi Teknis

Penggunaan elemen partisi seperti garis pemisah (----) dalam antarmuka berbasis teks menunjukkan bahwa mahasiswa telah mempertimbangkan aspek User Experience (UX). Dalam disiplin Teknik Komputer, kemampuan menyajikan informasi yang terstruktur adalah langkah awal menuju pengembangan antarmuka yang lebih kompleks. Estetika yang rapi memudahkan pengguna dalam navigasi menu dan meminimalisir kesalahan operasional. Hal ini membuktikan bahwa mahasiswa mulai memahami bahwa kode yang baik tidak hanya harus bekerja secara fungsional, tetapi juga harus komunikatif bagi penggunanya.

2. Implementasi Persistence melalui Struktur Perulangan

Keputusan 80% mahasiswa untuk menggunakan struktur `while True` dalam alur program menunjukkan pemahaman terhadap konsep program persistence. Dengan menerapkan perulangan tak terbatas yang dikendalikan oleh opsi keluar (exit), mahasiswa berhasil menciptakan aplikasi yang bersifat dinamis dan interaktif. Hal ini merupakan simulasi dari cara kerja perangkat lunak di dunia nyata yang tetap terjaga dalam memori sistem selama digunakan. Penggunaan struktur ini melatih mahasiswa dalam mengelola siklus hidup aplikasi (application lifecycle) secara sederhana namun efektif.

3. Validasi Logika dan Integritas Data

Penerapan validasi input (seperti memastikan data numerik pada input saldo) adalah capaian krusial dalam pengabdian ini. Mahasiswa belajar untuk tidak mempercayai input pengguna secara mentah-mentah (never trust user input), yang merupakan prinsip dasar keamanan dan ketahanan perangkat lunak. Meskipun validasi yang dilakukan masih



bersifat mendasar, keberhasilan mahasiswa dalam memaksa program untuk meminta input ulang saat terjadi kesalahan format menunjukkan bahwa mereka telah memahami konsep error handling. Kemampuan ini sangat penting untuk mencegah program mengalami terminasi mendadak (crash) akibat data yang tidak valid.

### **Refleksi dan Pengembangan Kompetensi**

Meskipun aplikasi yang dihasilkan masih bersifat prototype dan belum mencakup seluruh kompleksitas skenario dunia nyata, proses internalisasi yang terjadi melalui metode PBL ini telah memberikan fondasi profesionalisme koding yang kuat. Mahasiswa tidak lagi melihat pemrograman sebagai tugas matematika yang sekali selesai, melainkan sebagai sebuah proses membangun sistem yang harus dijaga kualitas dan daya tahannya. Bagi Program Studi Teknik Komputer, hasil ini menjadi bukti bahwa pengenalan standar koding sejak dini mampu mempercepat kesiapan mahasiswa dalam menempuh mata kuliah pengembangan perangkat lunak pada tingkat yang lebih lanjut.

## **5. Simpulan dan Saran**

Kegiatan pengabdian ini berhasil menginternalisasi prinsip Clean Code dan validasi logika pada mahasiswa Teknik Komputer melalui metode Problem-Based Learning (PBL). Hasil evaluasi menunjukkan bahwa seluruh mahasiswa mampu menghasilkan aplikasi berbasis Python yang fungsional dengan kualitas antarmuka (UI/UX) yang terstruktur. Penggunaan struktur perulangan persistence (while True) oleh 80% mahasiswa serta penerapan validasi tipe data pada seluruh proyek membuktikan adanya peningkatan disiplin teknis dalam membangun perangkat lunak yang reliabel. Mahasiswa tidak hanya menguasai sintaks pemrograman, tetapi juga memahami pentingnya ketahanan program (error handling) dan kenyamanan interaksi pengguna sebagai fondasi profesionalisme di bidang teknologi informasi.

Berdasarkan hasil yang dicapai, disarankan agar pada tahap selanjutnya mahasiswa mulai diperkenalkan dengan konsep penanganan eksepsi yang lebih kompleks (exception handling) dan modularitas kode menggunakan fungsi atau kelas (Object-Oriented Programming). Selain itu, integrasi pengembangan aplikasi dengan basis data eksternal perlu dipertimbangkan agar mahasiswa dapat mensimulasikan skenario dunia nyata yang lebih kompleks dan dinamis.

## **Referensi**

- Aitbayeva, B., Nurusheva, A., Mardenov, K., Yelubay, A., & Anuarbekov, A. (2023). Developing a curriculum for the "Clean Code" discipline in higher education institutions. *Eastern-European Journal of Enterprise Technologies*, 2(2 (122)), 14–24.
- Febriani, R., Irmayani, D., & Hendrawan, A. (2023). Peningkatan kemampuan pemecahan masalah dan hasil belajar siswa melalui model problem based learning pada mata pelajaran pemrograman dasar. *Jurnal Vokasi Informatika (JAVIT)*, 3(2), 80–87.
- Hermawan, H., Munir, M., & Wihardi, Y. (2021). Pengembangan media pembelajaran berbasis web untuk meningkatkan kemampuan clean code pada mata kuliah pemrograman berorientasi objek. *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, 7(1), 29–35.
- Indriasari, T. D., Luxton-Reilly, A., & Denny, P. (2020). A review of peer code review in higher education. *ACM Transactions on Computing Education (TOCE)*, 20(2), 1–34.
- Kurniawan, A., Adhisuwigno, S., & Sari, I. P. (2022). Analisis kesulitan mahasiswa dalam praktikum pemrograman dasar: Studi kasus pola pikir algoritmik dan sintaksis. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 9(3), 595–604.
- Rahman, A. A., Santosa, A. B., & Wijaya, I. K. S. (2024). Meningkatkan kualitas kode dan pemahaman konsep OOP melalui penerapan coding convention dan refactoring berbasis peer review. *Jurnal Teknik Informatika (Jutif)*, 5(1), 177–186.
- Rokhim, D. A., Rahmawati, N. I., & Asfani, K. (2023). Pengaruh peer review terhadap kualitas kode pemrograman dan kemampuan berpikir kritis mahasiswa. *Jurnal Edukasi dan*

- Penelitian Informatika (JEPIN), 9(2), 78–85.
- Sitanggang, N., Surbakti, A. H., & Sihombing, V. A. A. (2023). Penerapan model pembelajaran problem based learning untuk meningkatkan kemampuan pemrograman web siswa. *Journal of Informatics and Computer Science (JINACS)*, 2(1), 23–30.
- Tambunan, S. R., Sitorus, H., & Sinaga, E. R. (2022). Pengaruh model pembelajaran problem based learning terhadap hasil belajar pemrograman dasar. *Jurnal Teknologi Pendidikan: Jurnal Penelitian dan Pengembangan Pembelajaran*, 7(1), 131–138.
- Yuliana, Y., Nurhadi, D., & Suryani, N. (2024). Pengembangan kemampuan computational thinking melalui project-based learning dengan pendekatan scaffolding dalam pembelajaran pemrograman. *Edukatif: Jurnal Ilmu Pendidikan*, 6(1), 496–506.