

COMPARATIVE ANALYSIS OF NO-CODE AND CONVENTIONAL DEVELOPMENT EFFICIENCY IN BEAUTY PRODUCT APPLICATION REDESIGN

ANALISIS PERBANDINGAN EFISIENSI PENGEMBANGAN NO-CODE DAN KONVENSIONAL DALAM PERANCANGAN ULANG APLIKASI PRODUK KECANTIKAN

Hanum Zaqiah Permatasari¹, *Briansyah Setio Wiyono²

^{1,2}Universitas Muhammadiyah Malang, Jl. Raya Tlogomas No.246, Malang

hanumpermata@webmail.umm.ac.id¹, *brian@umm.ac.id²

* corresponding author

Abstract - Advances in information technology are driving the business sector to seek faster and more resource-efficient software development methods. Conventional methods such as PHP–MySQL often require longer development times and high technical expertise, while no-code approaches offer a simpler alternative for small to medium-scale applications. This study aims to analyze the efficiency of the no-code approach (MIT App Inventor) compared to conventional methods through a case study of the Innerlight application, applying the requirements engineering method that includes requirements formulation, analysis, specification, and validation of development results. Evaluation was conducted by measuring the processing time and effort estimation (person-days), as well as functional testing using black-box testing. The results show that no-code development requires an average of 6.5 working days (13 person-days), while the conventional method requires 8.5 working days (17 person-days), or approximately 24% more efficient in terms of time and effort. Efficiency was measured based on project observation data without financial estimation or analysis of variation between teams. This study is a single case study, so the results cannot be generalized to other projects of different scales and complexities. The no-code approach is considered suitable for simple applications, while conventional methods are superior for systems that require flexibility and advanced logic control.

Keywords - no-code development, conventional development, requirement engineering, black-box testing, effort estimation

Abstrak – Kemajuan teknologi informasi mendorong sektor bisnis untuk mencari metode pengembangan perangkat lunak yang lebih cepat dan efisien dalam penggunaan sumber daya. Metode konvensional seperti PHP–MySQL sering memerlukan waktu pengembangan yang lebih lama dan keahlian teknis tinggi, sementara pendekatan *no-code* menawarkan alternatif yang lebih sederhana untuk aplikasi berskala kecil hingga menengah. Penelitian ini bertujuan untuk menganalisis efisiensi pendekatan *no-code* (MIT App Inventor) dibandingkan metode konvensional melalui studi kasus aplikasi Innerlight, dengan menerapkan metode requirement engineering yang mencakup perumusan kebutuhan, analisis, spesifikasi, serta validasi hasil pengembangan. Evaluasi dilakukan dengan mengukur waktu pengerjaan dan estimasi upaya (*person-day*), serta pengujian fungsional menggunakan *black-box testing*. Hasil menunjukkan bahwa pengembangan dengan *no-code* membutuhkan rata-rata 6,5 hari kerja (13 *person-day*), sedangkan metode konvensional memerlukan 8,5 hari kerja (17 *person-day*), atau sekitar 24% lebih efisien dari segi waktu dan upaya kerja. Efisiensi diukur berdasarkan data observasi proyek tanpa estimasi finansial atau analisis variasi antar tim. Penelitian ini bersifat kasus tunggal, sehingga hasilnya belum dapat digeneralisasi ke proyek lain dengan skala dan kompleksitas berbeda. Pendekatan *no-code* dinilai sesuai untuk aplikasi sederhana, sedangkan metode konvensional lebih unggul untuk sistem yang membutuhkan fleksibilitas dan kontrol logika tingkat lanjut.

Kata Kunci – pengembangan *no-code*, pengembangan konvensional, requirement engineering, black-box testing, estimasi *person-day*

I. PENDAHULUAN

Teknologi informasi berkembang pesat dan mendorong berbagai sektor bisnis beralih ke era digital. Seiring perkembangan ini, permintaan terhadap aplikasi berbasis perangkat lunak meningkat, sehingga transformasi digital menuntut proses pengembangan yang cepat dan efisien. Namun, metode konvensional seperti PHP murni memiliki keterbatasan karena waktu pengembangan lebih lama dan sistem harus dibangun sepenuhnya dari awal tanpa dukungan framework[1]. Metode ini juga memerlukan debugging manual yang kompleks, sedangkan platform *no-code* menawarkan efisiensi dengan menghilangkan kebutuhan pengkodean berulang untuk aplikasi sederhana. Platform *no-code* memungkinkan pengguna tanpa kemampuan pemrograman membangun aplikasi melalui antarmuka visual[2][3]. Studi sebelumnya mengungkapkan bahwa platform *no-code* menghadapi kendala pada kustomisasi antarmuka, pengujian otomatis, dan banyaknya pertanyaan pengguna yang belum terselesaikan [4].

Munculnya platform *no-code* seperti MIT App Inventor menyederhanakan proses pengembangan aplikasi. Pengguna tanpa latar belakang teknis dapat membuat aplikasi lebih cepat dan efisien. Berdasarkan penelitian [5], metode *no-code* memberikan sekitar 90% hasil dari metode tradisional dengan hanya 20% biaya pengembangan, menandakan efisiensi signifikan pada dimensi biaya–kualitas. Platform ini mempercepat pembuatan aplikasi dengan meminimalkan waktu pengkodean sekaligus menekan upaya kerja. Selain waktu dan biaya, efisiensi juga dilihat dari kualitas hasil (defect), kemudahan pemeliharaan (maintainability), dan kecepatan perubahan (lead time change), di mana metode konvensional lebih fleksibel untuk perbaikan, sedangkan *no-code* lebih cepat namun terbatas pada pengujian dan pemeliharaan jangka panjang. Penggunaan *no-code* juga memperkuat kolaborasi antara tim bisnis dan IT, menjadikan organisasi lebih tanggap terhadap perubahan pasar dan kebutuhan pengguna [6]. Penelitian ini menggunakan pendekatan Requirement Engineering (RE) yang mencakup tahapan elicitation, analysis, specification, verification, dan validation untuk memastikan kebutuhan sistem terdokumentasi dan diimplementasikan dengan benar. Pengujian dilakukan dengan metode *black-box testing* untuk memverifikasi kesesuaian fungsi terhadap kebutuhan. Objek penelitian adalah aplikasi Innerlight dalam dua versi: konvensional berbasis PHP-XAMPP dan *no-code* menggunakan MIT App Inventor. Keduanya dibandingkan berdasarkan waktu pengembangan, ukuran tim, dan estimasi upaya kerja (*person-day*) sebagai indikator efisiensi.

Rumusan masalah penelitian ini menyoroti apakah terdapat perbedaan signifikan dalam efisiensi pengembangan antara metode *no-code* dan konvensional pada aplikasi sederhana seperti Innerlight. Berdasarkan hal tersebut, hipotesis yang diuji ialah: H_0 menyatakan tidak terdapat perbedaan signifikan (selisih *person-day* < 10%), sedangkan H_1 menyatakan metode *no-code* lebih efisien secara signifikan (selisih *person-day* \geq 10%). Tujuan penelitian ini adalah menguji secara kuantitatif tingkat efisiensi pengembangan antara kedua metode berdasarkan indikator waktu dan upaya kerja. Secara praktis, penelitian ini memberikan visualisasi hubungan antara waktu, personel, dan upaya kerja dalam satuan *person-day*, serta menawarkan alternatif pengembangan yang lebih efisien bagi tim dengan sumber daya terbatas, khususnya di sektor UMKM. Sebagai studi kasus, penelitian ini menggunakan aplikasi Innerlight, merek kecantikan lokal yang berjualan melalui marketplace. Dalam perencanaan proyek, digunakan ProjectLibre sebagai alat bantu manajemen karena bersifat gratis, mudah digunakan, dan memiliki fitur setara dengan Microsoft Project[7]. Pemanfaatan ProjectLibre memungkinkan perencanaan dan pengawasan proyek yang lebih terstruktur melalui penjadwalan tugas, alokasi sumber daya, serta Gantt Chart, sehingga mendukung pengelolaan waktu dan upaya kerja secara efektif.

II. SIGNIFIKANSI STUDI

A. Metode Konvensional

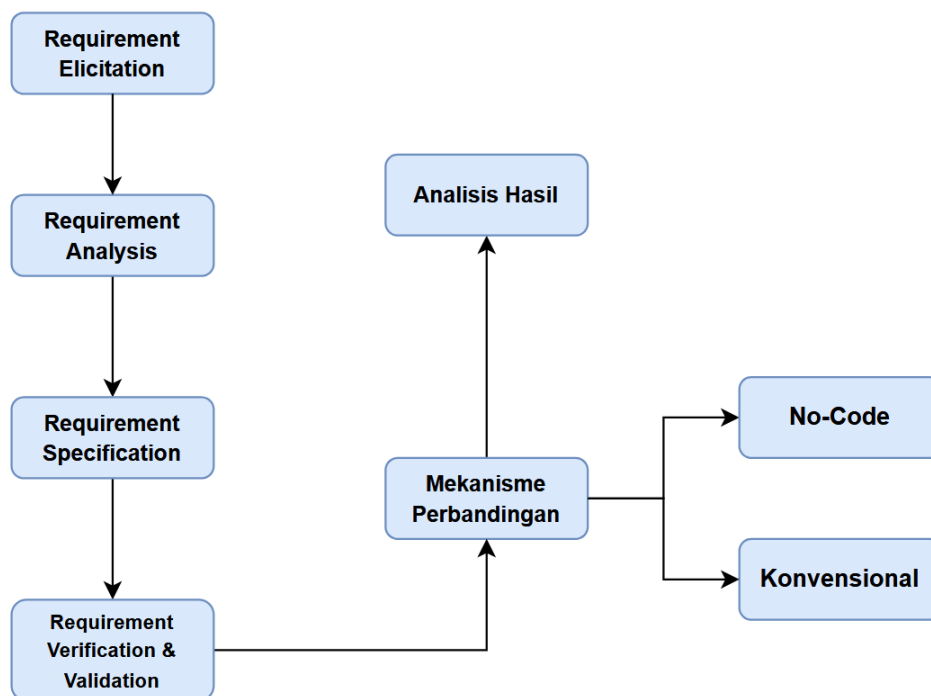
Metode konvensional seperti PHP–XAMPP memerlukan proses coding dan debugging manual, sehingga waktu pengembangannya lebih lama karena struktur sistem harus dibuat dari awal tanpa dukungan framework [8]. Penelitian lain juga menyebutkan bahwa penggunaan PHP murni membutuhkan waktu cukup panjang untuk proyek berskala menengah hingga besar [9]. Meski kurang efisien dalam waktu, metode ini unggul dalam performa. Penelitian [10] menunjukkan bahwa PHP–MySQL memiliki performa stabil dengan delay 46,134 ms, throughput 16,132 kbps, dan packet loss 0,00185528%, sehingga tetap relevan untuk sistem yang menuntut kestabilan dan kontrol penuh atas arsitektur aplikasi.

B. Metode No-Code

Pendekatan *no-code* menggunakan MIT App Inventor memungkinkan pembuatan aplikasi melalui antarmuka visual tanpa kemampuan coding tingkat lanjut. Metode ini mempercepat proses pengembangan dan cocok bagi pengguna non-teknis [11]. Riset menunjukkan MIT App Inventor banyak digunakan di sektor pendidikan dan bisnis untuk aplikasi berbasis daring [12], [13], menandakan meningkatnya minat terhadap solusi *no-code*. Penelitian [14] juga mencatat efektivitas 73–79% dalam aspek kualitas platform, kemudahan pemahaman, dan manfaat penggunaan, menegaskan potensi besar metode *no-code* dalam pengembangan aplikasi. Namun, pendekatan ini masih terbatas pada kustomisasi logika kompleks dan pengujian otomatis.

C. Requirement Engineering (RE)

Penelitian ini menggunakan pendekatan Requirement Engineering (RE) sebagai dasar perancangan ulang aplikasi Innerlight, yang mencakup proses sistematis mulai dari pengumpulan kebutuhan, validasi, hingga perbandingan antara metode konvensional (PHP–MySQL) dan *no-code* (MIT App Inventor) serta analisis hasilnya.



Gambar 1. Diagram Alur Metode Penelitian

Alur penelitian terdiri atas empat tahap utama, yaitu: (1) Requirement Elicitation, (2) Requirement Analysis, (3) Requirement Specification, dan (4) Requirement Verification & Validation [15], yang dijelaskan lebih lanjut bersama mekanisme perbandingan antara metode konvensional (PHP-MySQL) dan *no-code* (MIT App Inventor).

1. *Requirement Elicitation*, Penelitian ini diawali dengan pengumpulan kebutuhan fungsional dan non-fungsional yang dilakukan secara kolaboratif oleh tim pengembang, terdiri atas frontend developer (peneliti) dan backend developer (Shelly). Hasil identifikasi kebutuhan tersebut kemudian menjadi dasar dalam perancangan ulang antarmuka dan logika sistem pada dua pendekatan pengembangan, yaitu konvensional dan *no-code*.

Proses verifikasi kebutuhan dilakukan oleh observer eksternal (Ival) yang merupakan developer versi awal Innerlight. Observer berperan meninjau kesesuaian antara dokumen kebutuhan dan implementasi aktual tanpa mengetahui rincian teknis dari masing-masing pendekatan. Pendekatan ini dikategorikan sebagai *blinding parsial*, karena sebagian informasi dikendalikan agar penilaian tetap netral dan terfokus pada validitas kebutuhan sistem.

2. *Requirement Analysis*, Dalam diskusi tersebut, peneliti bertindak sebagai observer yang mendampingi skenario penggunaan untuk memetakan alur proses dari perspektif pengguna akhir. Peneliti melakukan wawancara terstruktur untuk memvalidasi sistem dengan mengajukan pertanyaan berikut:

- a. Fitur apa yang mana yang dapat disederhanakan tanpa mengurangi fungsi utama sistem?
- b. Fitur mana yang jarang digunakan atau justru menambah beban pemeliharaan?
- c. Apakah terdapat aspek dalam pengelolaan data produk atau alur checkout yang perlu diperbaiki?
- d. Jika ingin menerapkan mekanisme checkout internal, hambatan teknis atau operasional apa saja yang pernah dihadapi?

Hasil wawancara dicatat dan diverifikasi ulang untuk konfirmasi jawaban. Daftar kebutuhan akhir kemudian diklasifikasikan menjadi dua kategori: fitur utama (*must-have*) dan fitur pendukung (*nice-to-have*), yang selanjutnya divisualisasikan dalam bentuk use case diagram.

3. *Requirement Specification*, Pada tahap ini, kebutuhan sistem yang telah tervalidasi dikonversi menjadi spesifikasi formal berupa use case diagram yang menggambarkan hubungan antara pengguna dan fitur sistem. Use case ini menjadi dasar untuk wireframe, mockup, WBS, dan skenario pengujian black-box. Hasil pengamatan digunakan untuk menyesuaikan rancangan dengan kebutuhan pengguna. Keputusan penting meliputi:

- a. Penyederhanaan antarmuka agar lebih ringan dan cepat.
- b. Pengalihan proses checkout langsung ke WhatsApp menggantikan formulir berbasis database. Validasi kebutuhan dilakukan melalui diskusi daring dua hari dengan developer awal, dan hasilnya dijadikan acuan dalam perancangan arsitektur serta pengembangan antarmuka.



Gambar 2. Use Case Diagram

4. *Requirement Verification & Validation*, Uji coba aplikasi oleh pengguna, pengujian sistem dilakukan dengan black-box testing, yaitu menguji keluaran sistem berdasarkan input yang diberikan, tanpa mengecek logika program internal. Uji dilakukan pada kedua versi:

- a. Konvensional: Diuji melalui browser di server lokal, dicek CRUD produk, checkout form, validasi input.
- b. *No-code*: Diuji di emulator Android dan perangkat nyata, memeriksa stabilitas navigasi antar-screen, logika checkout ke WhatsApp, dan performa layout.

Pengujian fungsional menggunakan black-box testing memastikan seluruh fitur berfungsi sesuai kebutuhan pengguna. Pada pendekatan *no-code*, dilakukan 29 kasus uji mencakup enam modul (Home, Shop, Cart, Blog, Contact, About), dengan hasil 21 berhasil (72,4%) dan 8 gagal, terutama pada tombol “Selengkapnya” dan validasi kode kupon.

Pada metode konvensional, dilakukan 45 kasus uji dengan hasil 42 berhasil (93,3%) dan 3 gagal, umumnya pada integrasi form dan validasi input. Hasil ini menunjukkan kedua pendekatan memenuhi fungsi utama aplikasi, meski metode konvensional lebih stabil antar modul. Hasil pengujian digunakan untuk validasi akhir dan penyempurnaan prototipe, dengan pendampingan observer untuk memastikan kesesuaian keluaran dan ekspektasi. Revisi dilakukan hingga prototipe dinyatakan valid. Fokus perbaikan mencakup:

- a. Kesesuaian alur navigasi berdasarkan use case
- b. Keterbacaan dan konsistensi elemen antarmuka pengguna (UI)
- c. Stabilitas performa aplikasi pada perangkat berbeda
- d. Kinerja dan validasi form checkout

5. *Mekanisme Perbandingan*, Untuk memperoleh perbandingan objektif antara pendekatan konvensional dan *no-code*, seluruh kebutuhan sistem diuraikan menjadi unit tugas melalui Work Breakdown Structure (WBS). Setiap tugas mencakup aktivitas seperti analisis, perancangan antarmuka, pengembangan frontend-backend, integrasi, dan pengujian. Dua developer bekerja dengan peran seimbang: Hanum menangani desain dan navigasi, sedangkan Shelly fokus pada backend, validasi, dan integrasi checkout. Seluruh aktivitas dijadwalkan dan dimonitor menggunakan ProjectLibre, dengan jam kerja standar 08.00–17.00 WIB. Sistem secara otomatis menghitung durasi setiap aktivitas melalui fitur timesheet dan Gantt Chart, yang mencatat waktu mulai, selesai, pelaksana, serta jenis aktivitas. Data ini diverifikasi berdasarkan rekap harian dan bukti progres proyek seperti tangkapan layar, berkas APK, basis data, dan cuplikan kode. Visualisasi proyek disajikan dalam bentuk Gantt Chart untuk memperlihatkan urutan dan

ketergantungan antar-tugas. Semua aktivitas dicatat dalam Resource Sheet dan Gantt Chart ProjectLibre tanpa menggunakan alat manajemen proyek tambahan. Biaya dihitung secara simulatif berdasarkan total *person-day* dari hasil perhitungan ProjectLibre, dengan tarif asumsi IDR 50.000/jam sebagai acuan efisiensi waktu kerja. Perbandingan hasil meliputi jumlah *person-day*, durasi pengerjaan, dan prototipe akhir dari masing-masing pendekatan. Seluruh data WBS dianalisis lebih lanjut pada bab hasil untuk menilai efisiensi pengembangan antara metode konvensional dan no-code.

6. *Analisis Hasil*, Seluruh rangkaian pengembangan Innerlight didokumentasikan mencakup penjadwalan, dependensi antar-task, nama resource (Hanum & Shelly), serta perhitungan total *person-day* sebagai dasar pembandingan efisiensi metode.

1) Metode Konvensional

a. Analisis & Perancangan

- Use case & wireframe, hanum menganalisis kebutuhan berdasarkan evaluasi versi lama. Kegiatan ini mencakup diskusi mengenai kebutuhan dan permasalahan website sebelumnya, yang kemudian disesuaikan dengan dua metode utama.
- Mockup & prototype, hanum membuat mockup visual dan menggabungkan komponen wireframe ke dalam tata letak sambil mendokumentasikan pengaturan menu, navigasi dasar, dan struktur formulir checkout.

b. Front-end development, hanum mengembangkan halaman utama (Home, Shop, Blog, Contact Us, Cart) melalui pengembangan HTML, CSS, dan PHP, serta menerapkan desain responsif dan fitur navigasi.

c. Back-end development, shelly membangun basis data MySQL, koneksi PHP, dan fungsionalitas CRUD produk & blog. Ia juga menangani logika checkout dan validasi form.

d. Integrasi & pengujian, hanum dan shelly mengintegrasikan sistem, melakukan black-box testing, serta finalisasi konvensional di server lokal. Diskusi dilakukan untuk menyederhanakan halaman sebelum implementasi metode *no-code*.

2) Metode No-Code

a. Riset & Desain UI, hanum merancang ulang struktur navigasi pada MIT App Inventor, mendesain sembilan layar (screen) utama, serta memetakan jalur tombol checkout dan mempertimbangkan tata letak (layout) agar aplikasi tetap ringan dan optimal saat dijalankan di perangkat Android.

b. Pengembangan APP, shelly menyusun blok logika navigasi, membuat validasi input form checkout, merancang CRUD produk & viewer blog sederhana di MIT App Inventor, serta menautkan checkout ke WhatsApp.

c. Pengujian & Finalisasi, hanum & shelly bersama melakukan pengujian aplikasi di emulator Android, mencatat bug/error, merevisi blok logika jika diperlukan, hingga menghasilkan file APK final siap diuji di perangkat riil.

III. HASIL DAN PEMBAHASAN

1. Requirement Elicitation

Tahap Requirement Elicitation dilakukan untuk mengidentifikasi kebutuhan awal aplikasi Innerlight. Peneliti bersama developer backend mendokumentasikan fitur-fitur yang telah diterapkan sebelumnya, termasuk navigasi halaman, elemen produk, serta alur transaksi. Aktivitas ini tidak hanya bertujuan untuk melakukan inventarisasi fitur, tetapi juga untuk memahami konteks penggunaan sistem eksisting sebelum dilakukan redesain atau pengembangan ulang. Selain itu, dilakukan simulasi penggunaan sistem oleh developer awal dengan peneliti berperan sebagai observer. Simulasi penggunaan sistem membantu peneliti memahami perilaku pengguna yang sebenarnya saat menggunakan sistem sekaligus mengungkap potensi masalah interaksi yang dihadapi pengguna. Pembuatan prototipe mendukung proses ini karena memerlukan interaksi langsung dengan pengguna untuk mengungkap kebutuhan pengguna yang sebenarnya.

2. Requirement Analysis

Tahap Requirement Analysis bertujuan untuk mengolah hasil pengumpulan kebutuhan sehingga dapat dipilah menjadi kebutuhan utama dan kebutuhan pendukung. Proses validasi dilakukan secara bertahap, diawali dengan wawancara terstruktur, kemudian dilanjutkan melalui sesi diskusi daring bersama developer awal pada tanggal 23 Mei 2025. Tahapan ini dimaksudkan untuk memastikan kembali fitur yang benar-benar dibutuhkan oleh pengguna sekaligus mengidentifikasi fitur yang berpotensi disederhanakan atau dihilangkan. Berikut klasifikasi kebutuhan yang diperoleh berdasarkan hasil validasi:

- a. Fitur utama (*must-have*): katalog produk, sistem checkout, dan integrasi kontak WhatsApp.
- b. Fitur pendukung (*nice-to-have*): deskripsi produk lengkap, blog, contact, tampilan halaman depan yang kompleks.

3. Requirement Specification

Pemahaman kebutuhan pengguna dilakukan dengan meninjau katalog produk Innerlight di marketplace untuk menyesuaikan kebutuhan aktual dengan keterbatasan dua pendekatan, konvensional (PHP-MySQL) dan *no-code* (MIT App Inventor). Perubahan utama meliputi:

- a. Penyederhanaan detail produk agar tampilan layout lebih ringan dan responsif, terutama untuk mencegah potensi crash pada platform *no-code*.
- b. Penghapusan ikon keranjang belanja (cart icon) guna mengurangi kompleksitas visual yang tidak mendukung efisiensi tampilan antarmuka.
- c. Penyederhanaan form checkout, baik pada versi konvensional maupun *no-code*.
- d. Pengalihan proses checkout ke WhatsApp, menggantikan penyimpanan data ke database lokal pada versi *no-code*, untuk menyederhanakan alur transaksi.

Sinkronisasi antara kebutuhan awal dan perubahan dilakukan secara iteratif melalui klarifikasi yang berlangsung di grup chat. Hasil diskusi ini menghasilkan daftar kebutuhan akhir yang telah tervalidasi dan dapat dijadikan acuan dalam perancangan sistem. Daftar ini ditampilkan pada Tabel 1, dan digunakan sebagai dasar pengembangan prototipe, serta pembuatan task breakdown di tahap pengembangan berikutnya.

Tabel 1.
Daftar Kebutuhan Fitur Hasil Validasi

Fitur	Status Lama	Redesain	Catatan
Home	Ada	Redesain	Layout disederhanakan
Katalog Produk	Ada	Redesain	Detail produk dihilangkan
Blog	Ada	Redesain	Informasi ringkas saja
Keranjang Belanja	Ada	Redesain	Layout disederhanakan, ikon keranjang tidak diterapkan
Kontak	Ada	Redesain	Dialihkan ke WhatsApp
Check-out	Database	Redesain	Otomatis melalui database untuk konvensional solusi untuk <i>No-Code</i> melalui WhatsApp, dan penyimpanan ke database lokal

4. Requirement Verification & Validation

Validitas fungsional dinilai melalui pengujian fitur aplikasi Innerlight versi *no-code* (MIT App Inventor) dan konvensional (PHP-MySQL) berdasarkan skenario uji yang telah dirancang. Setiap skenario dikonversi menjadi test case terstruktur berisi ID, skenario, pre/post-condition, status, dan evidence link. Hasil pengujian (Tabel 2) menunjukkan versi *no-code* memiliki keberhasilan 72,4%, dengan beberapa kegagalan pada navigasi dan validasi input, sedangkan versi konvensional mencapai 93,3%, hanya mengalami kendala kecil pada form checkout yang kemudian diperbaiki. Tabel 2 menampilkan contoh test case utama tiap modul, sedangkan rincian lengkap tersedia dalam dokumen pengujian terpisah untuk verifikasi sistem.

Tabel 2.
Penguujian Versi Konvensional dan Penguujian Versi *No-Code*

Test Case ID	Test Scenario	Pre-Conditions	Post-Conditions	Status	Evidence link
Penguujian Versi Konvensional					
TC-HOME-001	Menguji tombol “Selengkapnya” di banner produk.	Aplikasi dijalankan melalui localhost (XAMPP).	Halaman detail produk terbuka.	Fail	https://github.com/hanumzaqiah/TESTCASE/blob/main/KONVENSIONAL-TC-HOME-0001.jpg
TC-BLOG-002	Akses halaman “Blog Innerlight”.	Browser aktif dan server lokal berjalan.	Detail artikel blog tampil.	Pass	https://github.com/hanumzaqiah/TESTCASE/tree/main/KONVENSIONAL-TC-BLOG-002
TC-SHOP-003	Fungsi “Add to Cart”.	Produk tersedia di halaman Shop.	Produk tampil di keranjang.	Pass	https://github.com/hanumzaqiah/TESTCASE/tree/main/KONVENSIONAL-TC-SHOP-003
TC-ABOUT US-004	Uji tampilan video promosi di halaman About Us.	Koneksi internet aktif.	Video dapat diputar.	Pass	https://github.com/hanumzaqiah/TESTCASE/blob/main/KONVENSIONAL-TC-ABOUT%20US-004.jpg
TC-CONTACT US-005	Form kendala/pertanyaan dengan form pesanan bisa langsung tersimpan ke database	Pengguna mengisi form pertanyaan/kendala	Form terkirim dan tersimpan di database	Pass	https://github.com/hanumzaqiah/TESTCASE/tree/main/KONVENSIONAL-TC-CONTACT%20US-005
TC-CART-006	Uji checkout saat produk kosong.	Cart kosong.	Tidak berpindah halaman, Cart berisi minimal 1 produk	Pass	https://github.com/hanumzaqiah/TESTCASE/tree/main/KONVENSIONAL-TC-CART-006
TC-FORMATT-007	Validasi input kosong pada billing details.	Pengguna menekan “Submit” tanpa mengisi form.	Form terkirim.	Pass	https://github.com/hanumzaqiah/TESTCASE/tree/main/KONVENSIONAL-TC-FORMAT-007
TC-FORMATT-008	Validasi input email tanpa format @.	Field di isi tanpa @	Pesan error muncul.	Pass	https://github.com/hanumzaqiah/TESTCASE/tree/main/KONVENSIONAL-TC-FORMAT-008
Penguujian Versi <i>No-Code</i>					
TC-HOME-001	Menguji tombol “Selengkapnya” pada banner produk.	Aplikasi dijalankan di emulator Android, halaman Home terbuka.	Pengguna berpindah ke halaman detail produk.	Fail	https://github.com/hanumzaqiah/TESTCASE/blob/main/TC-HOME-001.jpg
TC-BLOG-002	Menguji akses ke menu “Blog Innerlight”.	Aplikasi aktif, jaringan stabil.	Halaman blog tampil dengan daftar artikel.	Pass	https://github.com/hanumzaqiah/TESTCASE/tree/main/TC-BLOG-002
TC-SHOP-003	Verifikasi fungsi tombol “Add to Cart”.	Pengguna berada di halaman Shop.	Produk muncul di halaman Cart.	Pass	https://github.com/hanumzaqiah/TESTCASE/tree/main/TC-SHOP-003
TC-ABOUT US-004	Uji tampilan video promosi di halaman About Us.	Koneksi internet aktif.	Video dapat diputar.	Pass	https://github.com/hanumzaqiah/TESTCASE/blob/main/TC-ABOUT%20US-004.jpg
TC-CONTACT US-005	Form kendala/pertanyaan dengan form pesanan bisa langsung terkirim ke WhatsApp	Pengguna mengisi form pertanyaan/kendala	Form terkirim ke WhatsApp	Pass	https://github.com/hanumzaqiah/TESTCASE/tree/main/TC-CONTACT%20US-005

Test Case ID	Test Scenario	Pre-Conditions	Post-Conditions	Status	Evidence link	
TC-CART-006	Uji checkout saat keranjang kosong.	Cart kosong.	Muncul pesan cart berisi minimal 1 produk	Pass	https://github.com/hanumzaqiah/TESTCASE/tree/main/TC-CART-006	
TC-FORMA T-007	Validasi input kosong pada billing details.	Pengguna menekan "Submit" tanpa mengisi form.	Form dikirim.	gagal	Fail	https://github.com/hanumzaqiah/TESTCASE/tree/main/TC-FORMAT-007
TC-FORMA T-008	Validasi format nomor telepon.	Pengguna memasukkan huruf dalam kolom nomor.	Form dikirim.	gagal	Fail	https://github.com/hanumzaqiah/TESTCASE/tree/main/TC-FORMAT-008

Pengujian dilakukan secara manual dengan pendampingan langsung oleh peneliti terhadap sejumlah pengguna terbatas. Selama proses pengujian, setiap hasil observasi dicatat secara sistematis, dan setiap temuan bug segera ditindaklanjuti. Perbaikan dilakukan secara iteratif hingga sistem dinyatakan valid secara fungsional dan siap diimplementasikan. Hasil Pengujian:

- Sebagian besar fitur utama pada kedua versi aplikasi berhasil melewati pengujian dengan status *Pass*.
- Versi *no-code* (MIT App Inventor) menunjukkan performa baik pada perangkat menengah, meskipun terdapat *lag* minor pada perangkat dengan kapasitas RAM rendah.
- Beberapa kasus uji pada versi *no-code* mengalami kegagalan (*Fail*), terutama pada fitur navigasi tombol "Selengkapnya" dan validasi format input, yang disebabkan oleh keterbatasan kustomisasi dan kontrol logika pada platform MIT App Inventor.
- Tidak ditemukan bug kritis, namun beberapa penyesuaian antarmuka (UI) dilakukan berdasarkan masukan pengguna.
- Validasi input pada versi konvensional (PHP–MySQL) lebih fleksibel karena memiliki ruang kendali logika yang lebih luas dibandingkan dengan versi *no-code*.

Tahap evaluasi dan perbaikan fitur dilakukan untuk memastikan bahwa sistem yang dikembangkan dapat berjalan sesuai dengan kebutuhan pengguna dan tidak menimbulkan kendala saat digunakan. Evaluasi dilakukan secara manual melalui skenario berbasis test case dengan peneliti mengamati pengembang mengoperasikan aplikasi. Pengujian black-box adalah metode yang dipilih, yang menyelidiki fungsionalitas sistem, tanpa memeriksa kode yang mendasarinya. Pengujian berfokus pada aspek-aspek berikut:

- Kesesuaian alur navigasi antarmuka
- Keterbacaan dan konsistensi elemen UI.
- Stabilitas performa aplikasi pada berbagai perangkat.
- Kelancaran dan kepraktisan proses checkout.

Pengujian dilakukan baik pada versi konvensional (PHP–MySQL) maupun *no-code* (MIT App Inventor), dengan mencatat output aktual dibandingkan output yang diharapkan. Selama evaluasi, ditemukan beberapa kendala teknis yang kemudian diperbaiki secara iteratif. Peneliti dan pengembang tetap berkomunikasi melalui grup obrolan bersama untuk membahas solusi dan mengirimkan pembaruan. Temuan utama dan penyempurnaan lanjutan disajikan dalam tabel berikut:

Tabel 3.
Evaluasi dan Revisi

No	Aspek Evaluasi	Hasil Temuan	Perbaikan yang Dilakukan
1	Navigasi Menu	Beberapa tombol tidak responsif di perangkat	Penyesuaian posisi dan fungsi tombol
2	Ukuran Gambar	Gambar beresolusi tinggi memperlambat loading	Kompresi ukuran gambar tanpa menurunkan kualitas
3	Alur Transaksi	Checkout di <i>no-code</i> belum optimal	Checkout diarahkan langsung ke WhatsApp
4	Layout Tampilan	Layout penuh menyebabkan aplikasi crash pada <i>no-code</i>	Elemen non-prioritas dihapus
5	Data Transaksi	Perbedaan penyimpanan data antara dua versi	Validasi alur checkout di masing-masing versi

Siklus evaluasi dan revisi ini diulang hingga kedua versi menunjukkan performa yang stabil dan user-friendly. Perbaikan fokus pada aspek yang paling berpengaruh terhadap kenyamanan pengguna, seperti kecepatan akses, kesederhanaan alur, dan kompatibilitas tampilan. Hasil akhir dari tahap ini menunjukkan bahwa metode konvensional maupun *no-code* telah mencapai level kesiapan implementasi yang layak, dengan risiko kesalahan sistem yang minimal.

5. Mekanisme Perbandingan

Untuk memastikan perbandingan antara pendekatan konvensional dan *no-code* dilakukan secara objektif, seluruh kebutuhan sistem dijabarkan ke dalam unit tugas detail melalui Work Breakdown Structure (WBS). Setiap tugas mencakup analisis kebutuhan, desain antarmuka, pengembangan frontend-backend, integrasi, pengujian, dan finalisasi prototipe. Hanum berfokus pada desain dan navigasi, sedangkan Shelly menangani logika backend, validasi, dan integrasi checkout.

Seluruh aktivitas dijadwalkan menggunakan ProjectLibre dengan jam kerja standar 08.00–17.00 WIB dan durasi minimum 0,5 hari. ProjectLibre menghitung otomatis waktu, ketergantungan antar-tugas, serta total durasi proyek melalui fitur Gantt Chart dan Resource Sheet. Data timesheet harian dimasukkan langsung ke ProjectLibre tanpa menggunakan alat tambahan, menghasilkan perhitungan yang konsisten dan terukur. Gantt Chart digunakan sebagai visualisasi utama perbandingan, menampilkan urutan aktivitas, durasi tugas, serta hubungan antar-proses. Nilai *person-day* dihitung otomatis oleh ProjectLibre dan digunakan untuk estimasi biaya serta analisis efisiensi waktu.

Name	Duration	Start	Finish	Predecessors	Resource Names
INNERLIGHT	15.5 days?	2025-05-23 08:00	2025-06-13 13:00		
Metode Konvensional	8.5 days?	2025-05-23 08:00	2025-06-04 13:00		
Analisis & Perancangan	2 days?	2025-05-23 08:00	2025-05-26 17:00		
Use Case & Wireframe	1 day?	2025-05-23 08:00	2025-05-23 17:00		Hanum
Mockup & Prototype	1 day?	2025-05-26 08:00	2025-05-26 17:00	4	Hanum
Frontend Development	3 days?	2025-05-27 08:00	2025-05-29 17:00		
Halaman Home, Shop, About Us	1 day?	2025-05-27 08:00	2025-05-27 17:00	5	Hanum
Halaman Blog, Artikel Blog, Contact, Cart, Checko	1 day?	2025-05-28 08:00	2025-05-28 17:00	7	Hanum
Styling Global, Navigasi, Responsif	1 day?	2025-05-29 08:00	2025-05-29 17:00	8	Hanum
Backend Development	3 days?	2025-05-30 08:00	2025-06-03 17:00		
Setup DB & Koneksi PHP-MySQL	1 day?	2025-05-30 08:00	2025-05-30 17:00	9	Shelly
CRUD Produk & Artikel Blog	1 day?	2025-06-02 08:00	2025-06-02 17:00	11	Shelly
Checkout Logic & Validasi	1 day?	2025-06-03 08:00	2025-06-03 17:00	12	Shelly
Integrasi & Pengujian	0.5 days?	2025-06-04 08:00	2025-06-04 13:00		
Integrasi & Final Testing	0.5 days?	2025-06-04 08:00	2025-06-04 13:00	13	Hanum;Shelly
Metode No-Code	6.5 days?	2025-06-05 08:00	2025-06-13 13:00		
Riset & Desain UI	2 days?	2025-06-05 08:00	2025-06-06 17:00		
Riset Struktur Navigasi	1 day?	2025-06-05 08:00	2025-06-05 17:00		Hanum

Gambar 3. Gantt Chart

Dalam penelitian ini, estimasi effort dihitung menggunakan satuan *person-day*, yaitu total beban kerja untuk menyelesaikan proyek dengan mengalikan jumlah personel dengan jumlah hari kerja yang dibutuhkan:

$$Person\text{-}Day = \text{Jumlah Personel} \times \text{Durasi Hari Kerja} \quad (1)$$

$$\begin{aligned} \text{Efisiensi} &= (\text{Effort konvensional} - \text{Effort no-code}) / \text{Effort konvensional} \times 100\% \quad (2) \\ &= (17 - 13) / 17 \times 100\% = 4 / 17 \times 100\% \approx 23.53\% \end{aligned}$$

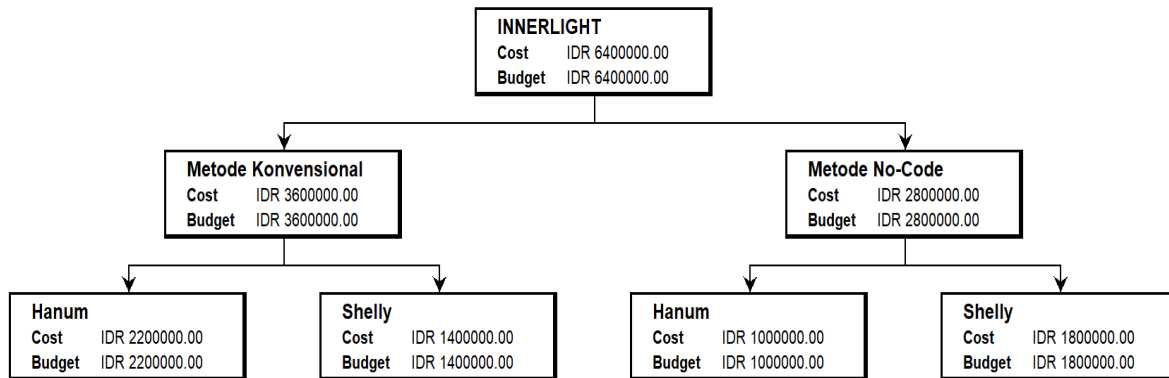
Walaupun jumlah personel yang terlibat sama (dua orang), terdapat perbedaan durasi pengerjaan antara kedua metode. Berdasarkan simulasi ProjectLibre, pendekatan konvensional (PHP dan MySQL) memerlukan 8,5 hari kerja, sedangkan pendekatan *no-code* (MIT App Inventor) hanya membutuhkan 6,5 hari kerja, menghasilkan estimasi efisiensi sekitar 24%. Nilai ini bersifat deskriptif, karena penelitian hanya melibatkan satu proyek dan satu tim pengembang, sehingga tidak dilakukan uji statistik inferensial seperti confidence interval (CI) atau bootstrap untuk menguji signifikansi perbedaannya. Oleh sebab itu, hasil ini dipandang sebagai indikasi empiris efisiensi waktu, bukan generalisasi statistik.

Tabel 4.
Effort Estimation Based on Development Method

Development Method	Development Duration (Days)	Personnel	Estimated Effort (Person-Days)
Conventional (PHP and MySQL)	8,5	2	17
No Code (MIT App Inventor)	6,5	2	13

6. Analisis Hasil

Seluruh proses pengembangan aplikasi Innerlight didokumentasikan secara menyeluruh untuk memastikan keterlacakan proses dan akurasi perbandingan antara metode. Dokumentasi ini mencakup perencanaan proyek, alokasi tugas, penjadwalan berbasis Gantt Chart, penghitungan effort *person-day*, serta output tiap tahapan pengembangan. Data dokumentasi ini disusun berdasarkan log aktivitas selama proyek berlangsung dan divalidasi dengan hasil observasi serta pengujian sistem.



Gambar 4. Resource Sheet

Pendekatan konvensional memerlukan 17 *person-day*, sedangkan *no-code* hanya 13 *person-day*. Dokumentasi mencakup Resource Sheet berisi peran, aktivitas, dan beban kerja tiap anggota untuk mempermudah alokasi tugas. Hasil black-box testing menunjukkan semua fitur utama berfungsi, meski versi *no-code* mengalami beberapa kegagalan karena keterbatasan logika blok. Versi konvensional lebih fleksibel dalam validasi input berkat kontrol kode yang lebih luas. Efisiensi MIT App Inventor didukung oleh blok visual dan komponen drag-and-drop yang mempercepat pembuatan antarmuka, namun terbatas dalam kustomisasi kompleks, manajemen data besar, dan integrasi CI/CD. Sebaliknya, metode konvensional mendukung CI/CD otomatis melalui GitHub Actions, yang mempermudah proses build, testing, dan deployment.

KESIMPULAN

Berdasarkan Hasil pengujian menunjukkan bahwa pendekatan *no-code* (MIT App Inventor) lebih efisien dibanding konvensional (PHP–MySQL) dalam proyek Innerlight. Pengembangan dengan *no-code* memerlukan 6,5 hari (13 *person-day*), sedangkan metode konvensional membutuhkan 8,5 hari (17 *person-day*). Estimasi ini menggambarkan efisiensi waktu, bukan analisis biaya. Kedua metode memenuhi seluruh kebutuhan fungsional utama aplikasi (Home, Blog, Contact, About, dan Cart). MIT App Inventor unggul dalam kecepatan berkat blok visual dan komponen drag-and-drop, namun terbatas dalam kustomisasi kompleks dan integrasi CI/CD. Sementara itu, PHP–MySQL lebih fleksibel dan stabil meski memerlukan waktu lebih lama. Dengan demikian, *no-code* cocok untuk proyek cepat dan sederhana, sedangkan metode konvensional lebih sesuai untuk sistem kompleks yang membutuhkan stabilitas dan kontrol penuh terhadap kode.

REFERENSI

- [1] R. Khankhoje, "Beyond Coding: A Comprehensive Study of Low-Code, No-Code and Traditional Automation," *J. Artif. Intell. Cloud Comput.*, vol. 1, no. 4, pp. 1–5, 2022.
- [2] D. Novita, A. Farisi, and F. P. Sihotang, "Pembuatan Aplikasi Mobile Tanpa Coding Bagi Karyawan PT Dunia Kimia Utama," *Fordicate*, vol. 2, no. 1, pp. 29–39, 2022.
- [3] S. F. A. Razak, Y. P. Ernn, F. I. Yussoff, U. A. Bukar, and S. Yogarayan, "Enhancing Business Efficiency through Low-Code/No-Code Technology Adoption: Insights from an Extended UTAUT Model," *J. Human, Earth, Futur.*, vol. 5, no. 1, pp. 85–99, 2024.
- [4] K. Żyła, K. Chwaleba, and D. Choma, "Evaluating Usability and Accessibility of Visual Programming Tools for Novice Programmers—The Case of App Inventor, Scratch, and StarLogo," *Appl. Sci.*, vol. 14, no. 21, pp. 1–16, 2024.
- [5] S. Y. Simhadri, "No-Code vs Traditional Machine Learning for Lead Generation: A Comparative Case Study," *Int. J. Emerg. Trends Comput. Sci. Inf. Technol.*, vol. ICCSAIML-2, pp. 118–123, 2025.
- [6] G. Masili, "No-code Development Platforms: Breaking the Boundaries between IT and Business Experts," *Int. J. Econ. Behav.*, vol. 13, no. 1, pp. 33–49, 2023.
- [7] A. T. Oktaga, M. A. Heses, and K. Nurdianto, "Studi Literatur: Alternatif Pengganti Microsoft Project dalam Proses Penjadwalan Proyek," *J. Cakrawala Inf.*, vol. 2, no. 2, pp. 51–55, 2022.
- [8] A. Niarman, Iswandi, and A. K. Candri, "Comparative Analysis of PHP Frameworks for Development of Academic Information System Using Load and Stress Testing," *Int. J. Softw. Eng. Comput. Sci.*, vol. 3, no. 3, pp. 424–436, 2023.
- [9] Robby Yuli Endra, Yuthsi Aprilinda, Yanuaris Yanu Dharmawan, and Wahyu Ramadhan, "Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website," *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 11, no. 1, pp. 48–55, 2021.
- [10] E. A. Putri, U. Latifa, and A. Stefanie, "Web System for Data Collection and Data Transmission with MySQL Data Storage," *JISICOM (Journal Inf. Syst. Informatics Comput.)*, vol. 6, no. 2, pp. 343–360, 2022.
- [11] A. Kholik, A. Soegiarto, and S. R. Putri, "Perancangan Aplikasi Sistem Peminjaman Laboratorium Berbasis No-Code Development Platforms (NCDPs) Menggunakan Glideapps Abdul," *J. Ilm. Multidisiplin*, vol. 3, no. 1, pp. 285–296, 2024.
- [12] A. Mursyid and S. Stanley, "Developing E-dictionary based on MIT App Inventor for Preschool Students," *Indones. J. Multidiscip. Educ. Res.*, vol. 2, no. 2, pp. 191–205, 2024.
- [13] J. Hartono, M. Vania, L. Cahyono, and M. Azzawan, "Pengembangan Katalog Online Kue Menggunakan MIT App Inventor dengan Metode RAD," *J. Inform. dan Sist. Inf.*, vol. 8, no. 2, pp. 41–49, 2023.
- [14] M. H. Putri, M. Jannah, I. Fitriati, and R. Ridwan, "Efektivitas Penggunaan Mit App Inventor Pada STKIP Taman Siswa Bima," *J. Sist. dan Teknol. Inf.*, vol. 12, no. 2, pp. 351–356, 2024.
- [15] R. Sari, A. Muhammad, M. Salimy, M. I. Arief, and M. Rezza, "Understanding of Requirements Engineering using The Three Dimensions of Requirements Engineering Method in Platform Development," vol. 5, no. 2, pp. 99–111, 2023.