

Penerapan Clean Arsitektur Menggunakan Mobile Platform Berbasis Flutter Untuk Aplikasi Layanan Kapal

^{1*}Amanda Putri, ²Jefry Sunupurwa Asri, ³Nizirwan Anwar, ⁴Adi Widiatono
¹²³⁴Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Esa Unggul,
DKI Jakarta

E-mail: ¹amandaputri.ap23@student.esaunggul.ac.id (Corresponding Author),
²jefry.sunupurwa@esaunggul.ac.id, ³nizirwan.anwar@esaunggul.ac.id,
⁴adi.widiatono@esaunggul.ac.id

ABSTRAK

Tingkat adopsi teknologi mobile di Indonesia mencapai 77,02% memberikan peluang besar bagi perusahaan untuk meningkatkan layanan mereka melalui platform mobile. PT ABC, penyedia layanan kapal untuk berbagai kebutuhan, saat ini hanya menggunakan aplikasi berbasis web dan membutuhkan *aplikasi layanan kapal* berbasis mobile untuk meningkatkan efisiensi dan fleksibilitas. Penelitian ini bertujuan untuk menerapkan *Clean Architecture* pada platform mobile berbasis Flutter untuk memperbaiki struktur dan memudahkan pemeliharaan kode dalam *aplikasi layanan kapal*. Metode penelitian yang digunakan adalah metode prototyping, dengan tahapan yang meliputi Communication, Quick Plan, Modelling Quick Design, Pembentukan *Prototype* dan Deployment Delivery & Feedback. Pendekatan ini memungkinkan pengembangan aplikasi yang lebih cepat, identifikasi potensi perubahan lebih awal, dan pengurangan risiko kegagalan atau ketidaksesuaian dengan kebutuhan pengguna. Hasil penelitian menunjukkan bahwa penerapan *Clean Architecture* dalam *aplikasi layanan kapal* berbasis mobile berhasil diimplementasikan dengan baik sehingga memberikan kemudahan dalam pemeliharaan. Pengujian Blackbox dengan 10 kasus uji memastikan bahwa aplikasi berfungsi sesuai dengan harapan dan siap digunakan oleh PT ABC. User Acceptance Testing juga menunjukkan bahwa aplikasi ini memenuhi kebutuhan pengguna dan meningkatkan responsivitas perusahaan terhadap perubahan pasar, hasil pengujian mencapai 86,83%. Dengan pembagian logika bisnis ke dalam folder usecases, efisiensi pengembangan, fleksibilitas perubahan, dan kemudahan pemeliharaan sistem secara keseluruhan meningkat. Kesimpulannya, *Clean Architecture* terbukti sebagai solusi yang tepat untuk mengoptimalkan operasi bisnis PT ABC, memudahkan pemeliharaan dan pengujian aplikasi, serta meningkatkan pemahaman dan keterbacaan kode bagi pengembang.

Kata kunci : *Aplikasi Layanan Kapal, Clean Arsitektur, Prototype.*

ABSTRACT

Indonesia's mobile technology adoption rate of 77.02% provides a great opportunity for companies to improve their services through mobile platforms. PT ABC, a ship service provider for various needs, currently only uses web-based applications and needs a mobile-based ship service application to increase efficiency and flexibility. This research aims to implement Clean Architecture on a Flutter-based mobile platform to improve the structure and facilitate code maintenance in ship service applications. The research method used is the prototyping method, with stages including Communication, Quick Plan, Modelling Quick Design, Prototype Formation and Deployment Delivery & Feedback. This approach enables faster application development, earlier identification of potential changes, and reduced risk of failure or mismatch with user needs. The results showed that the application of Clean Architecture in mobile-based ship service

applications was successfully implemented so that it provided ease of maintenance. Blackbox testing with 10 test cases ensures that the application functions as expected and is ready to be used by PT ABC. User Acceptance Testing also shows that this application meets the needs of users and increases the company's responsiveness to market changes, the test results reached 86.83%. With the division of business logic into usecases folders, the overall efficiency of development, flexibility of change, and ease of maintenance of the system is increased. In conclusion, Clean Architecture proved to be the right solution to optimise PT ABC's business operations, facilitate application maintenance and testing, and improve code understanding and readability for developers.

Keyword : Ship Service, Application, Clean Architecture, Prototype.

1. PENDAHULUAN

Menurut APJII, jumlah pengguna internet di Indonesia terus meningkat dan akan mencapai 210 juta orang, atau 77,02% dari total populasi, pada 2021. Angka ini naik dari 64,80 persen pada 2018 dan 73,70 persen pada 2019 dan 2020, menunjukkan bahwa internet sekarang menjadi kebutuhan dasar bagi banyak orang (Asosiasi Penyelenggara Jasa Internet Indonesia, 2022). Lebih dari 89% pengguna internet Indonesia menggunakan perangkat mobile pada tahun 2021, menurut Kementerian Komunikasi dan Informatika (Adisty, 2022).

PT ABC ingin mengembangkan aplikasi layanan kapal berbasis mobile yang mudah dikelola, diuji, dan dipelihara untuk memenuhi kebutuhan pengguna yang berkembang. Clean Architecture dianggap sebagai solusi untuk struktur dan pemeliharaan kode aplikasi guna meningkatkan maintainability (Abdillah, 2024).

Penelitian ini akan menggunakan metode prototyping untuk merancang aplikasi layanan kapal, dengan penerapan Clean Architecture. Metode ini dipilih untuk mengatasi kekurangan penelitian terdahulu seperti kurangnya fitur detail dan masalah pada metode waterfall, serta untuk meningkatkan keterlibatan pengguna dan fleksibilitas pengembangan (Hidayatulloh, 2021).

Aplikasi layanan kapal dirancang menggunakan Clean Architecture pada platform mobile berbasis Flutter dan bahasa pemrograman Dart. Penerapan

arsitektur ini memungkinkan pemisahan komponen-komponen aplikasi, sehingga mempermudah pemeliharaan dan perbaikan (Badrudduja, 2022). Penerapan Clean Architecture dalam pengembangan aplikasi layanan kapal berbasis mobile dengan Flutter diharapkan dapat meningkatkan efektivitas dan kenyamanan layanan. Dengan struktur yang jelas dan mudah dipelihara, aplikasi ini akan lebih mudah untuk dikembangkan dan diadaptasi sesuai kebutuhan masa depan.

2. LANDASAN TEORI

2.1 Clean Architecture

Penelitian oleh (Saifulloh, 2023) menunjukkan bahwa Clean Architecture menggabungkan berbagai arsitektur untuk menciptakan sistem yang mudah diuji dan tidak bergantung pada framework atau basis data tertentu. Arsitektur ini membagi aplikasi menjadi tiga lapisan: domain layer untuk logika bisnis, data layer untuk mengelola data, dan presentation layer untuk menampilkan data.

2.2 Mobile

Mobile adalah program yang dirancang khusus untuk membantu pengguna mencapai tujuan tertentu sesuai dengan keunggulan dan kemampuan aplikasinya (Devina Vania, 2022).

2.3 Platform

Platform adalah wadah digital yang digunakan untuk menjalankan sistem sesuai program yang telah dibuat, terdiri dari sekumpulan program yang dapat

dioperasikan pada PC atau Android (Eka Yuda Wibawa, 2021).

2.4 Aplikasi

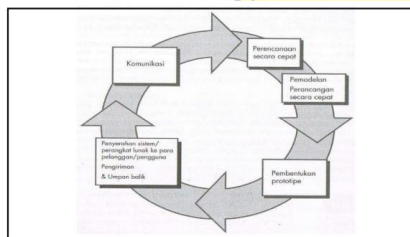
Menurut Jogiyanto, aplikasi adalah perangkat lunak dengan berbagai fitur untuk mengolah data, menampilkan informasi, dan memudahkan pengguna melakukan berbagai tugas sesuai kebutuhan (Kinaswara, 2019).

2.5 Android

Android adalah sistem operasi seluler modern yang mendukung pengembangan aplikasi pada perangkat canggih, dengan prioritas pada aplikasi asli dan kontrol terhadap distribusi aplikasi pihak ketiga (Nelly Sofi & Riza Dharmawan, 2022).

2.6 Prototype

Metode prototyping dalam pengembangan perangkat lunak membuat versi awal sistem yang memungkinkan interaksi pengembang dan pengguna, dengan prototipe yang disempurnakan berdasarkan pengujian dan evaluasi selama proses pengembangan (Silvi Purnia, 2021).



Gambar 1. Metode Prototype (Margareta & Putra, 2022)

Langkah-langkah metode prototype meliputi Communication dan pengumpulan data awal mencakup analisis kebutuhan pengguna, Quick Plan yang merupakan tahapan perencanaan kebutuhan, Modelling Quick Design merupakan tahapan pembuatan desain, Pembentukan Prototype merupakan tahapan pembuatan perangkat prototype baik itu pengujian maupun penyempurnaan, Deployment Delivery &

Feedback, yang merupakan tahap mengevaluasi prototype dan memperhalus analisis untuk memenuhi kebutuhan pengguna (Aditya, 2021).

2.7 Flutter

Flutter adalah SDK open-source dari Google untuk pengembangan aplikasi seluler dan sistem operasi Google Fuchsia. Dibangun dengan Skia Graphics Engine, Flutter menggunakan widget yang dapat disesuaikan seperti Material Design dan Cupertino untuk membuat antarmuka asli dengan cepat di Android dan iOS (Nelly Sofi, 2022).

2.8 Strapi

Strapi adalah CMS Headless open-source yang memungkinkan editor mengelola konten melalui panel admin, sementara pengembang bebas menggunakan alat pilihan mereka. Strapi memiliki panel admin yang dapat diperluas, API fleksibel, dan sistem pengguna bawaan untuk mengatur akses informasi (Alhazmy, 2023).

2.9 UML (Unified Modeling Language)

UML adalah bahasa pemodelan berbasis objek untuk menspesifikasi, memvisualisasi, dan mendokumentasikan sistem perangkat lunak. UML digunakan untuk memodelkan berbagai jenis sistem di berbagai bidang (Prasetya, 2022).

2.10 Black-box Testing

Black box testing adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi berdasarkan spesifikasi tanpa melihat struktur internalnya, dengan mengidentifikasi input dan output yang diharapkan (Putra, 2020).

2.11 User Acceptance Testing

UAT menguji interaksi langsung antara sistem dan pengguna untuk memastikan bahwa fitur berjalan sesuai dengan kebutuhan pengguna

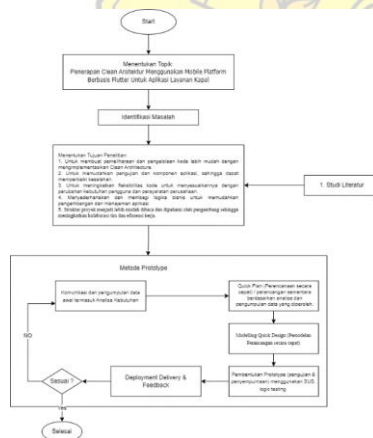
(Chamida, 2021). Rumus presentase sebagai berikut :

$$P = \frac{s}{skor} \times 100\% \text{ [i]}$$

Keterangan : P (Presentase), Jumlah frekuensi dikali dengan skor (s), skor tertinggi dikali dengan jumlah ideal sampel (skor).

3. METODOLOGI

Proses pengembangan sistem melibatkan penerapan metode Prototype, yang terdiri dari lima tahap desain utama, yakni Communication, Quick Plan, Modelling Quick Design, Pembentukan Prototype, Deployment Delivery & Feedback. Berikut adalah langkah-langkah yang diambil dalam proses pengembangan sistem:



Gambar 2. Kerangka Berpikir

Berdasarkan kerangka berpikir dapat dijelaskan bahwa:

1. Menentukan Topik

Penulis menentukan topik penelitian dan judul serta kebutuhan yang diperlukan dalam pengerjaan tugas akhir ini.

2. Identifikasi Masalah

Tahap identifikasi masalah merupakan langkah penting dalam penelitian, di mana peneliti

mengidentifikasi isu atau kebutuhan yang perlu diselesaikan atau dijawab oleh penelitian.

3. Studi Literature

Pada tahap ini peneliti melakukan pengumpulan data untuk kebutuhan penelitian yang dimulai dari jurnal, artikel yang dijadikan sebagai referensi untuk penelitian.

4. Communication

Pengembang dan user bertemu dan berbicara satu sama lain untuk menentukan tujuan penelitian pengumpulan data awal termasuk analisis kebutuhan.

5. Quick Plan

Melakukan perencanaan secara cepat berdasarkan analisa dan pengumpulan data yang diperoleh. Tahap ini menguraikan analisis kebutuhan system.

6. Modelling Quick Design

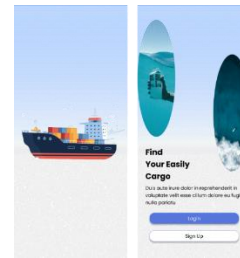
Memperoleh hasil analisis kebutuhan yang menyeluruh, dibuat desain prototipe awal dalam bentuk diagram Unifield Modeling Language (UML). UML merinci interaksi dan struktur elemen sistem.

7. Pembentukan Prototype

untuk menggambarkan secara visual struktur aplikasi layanan kapal berbasis Android. Dalam implementasi aplikasi, diharapkan prototipe ini dapat memberikan gambaran yang jelas dan terstruktur tentang bagaimana berbagai komponen dan fitur akan berinteraksi satu sama lain.

8. Deployment Delivery & Feedback

Pengujian dilakukan untuk memastikan bahwa aplikasi berjalan dengan baik sesuai dengan kebutuhan. Proses pengujian melibatkan dua metode yaitu Blackbox Testing dan User Acceptance Testing (UAT).



Gambar 3. Splash Screen dan Welcome

4. HASIL DAN PEMBAHASAN

4.1 Communication

Penelitian ini menggunakan dua metode pengumpulan data yaitu observasi dan wawancara.

1. Observasi

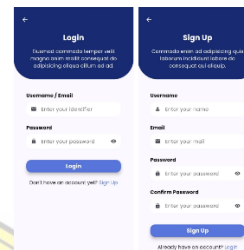
Observasi dilakukan untuk memahami proses layanan kapal di PT ABC dan mengidentifikasi kendala serta kebutuhan bisnis, termasuk manajemen pengguna, pemesanan layanan, dan riwayat pemesanan.

2. Wawancara

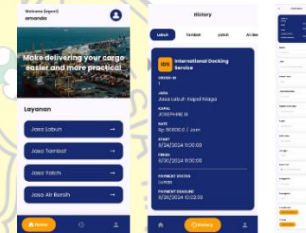
Wawancara dilakukan dengan responden yang memahami proses layanan kapal di PT ABC untuk mengumpulkan informasi rinci mengenai jenis layanan, prosedur pemilihan jasa, verifikasi dokumen, persetujuan, pembuatan invoice, dan kendala yang dihadapi.

4.2 Quick Plan

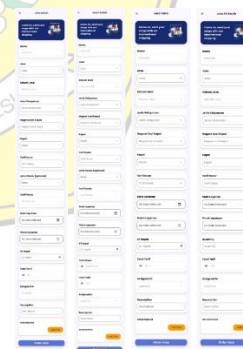
Pada tahap ini melakukan perencanaan dengan cepat berdasarkan data yang diperoleh dari observasi dan wawancara.



Gambar 4. Login dan Sign Up



Gambar 5. Home, History dan Detail History



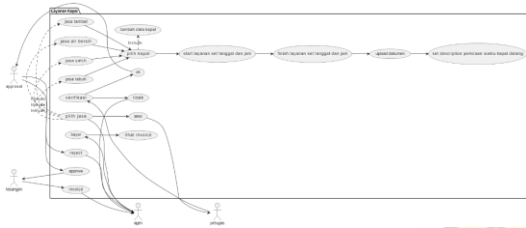
Gambar 6. Pemesanan Jasa Labuh, Tambat, Yatch dan Air Bersih

4.3 Modelling Quick Design

Pada tahap ini dibuat desain prototipe awal dalam bentuk diagram Unified Modeling Language (UML). Proses bisnis yang diusulkan dijelaskan melalui use case diagram, Activity Diagram dan Class Diagram.

1. Use Case Diagram

Use case diagram adalah jenis diagram UML yang menggambarkan hubungan antara sistem dan aktor, serta berbagai interaksi yang dilakukan pengguna dengan sistem (Prasetya, 2022).



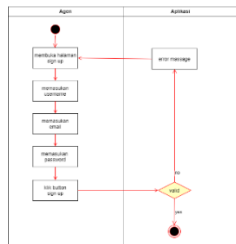
Gambar 6. Use Case Diagram

Pada aplikasi ini tidak ada role untuk admin karena data diambil dari inaportnet, Role pada aplikasi layanan kapal yaitu agent, operator, approval dan keuangan.

2. Activity Diagram

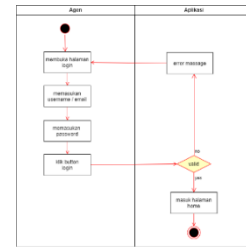
Activity Diagram adalah diagram UML yang mensimulasikan proses dalam sistem secara vertikal, menunjukkan rangkaian aktivitas atau alur yang berkembang dari Use Case, seperti menu atau proses bisnis dalam sistem (Prasetya, 2022).

1. Register



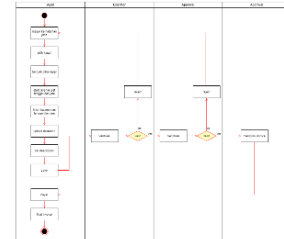
Gambar 7. Usecase Register

2. Login



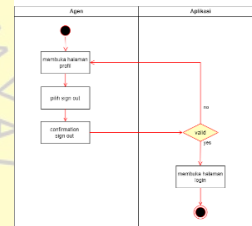
Gambar 8. Usecase Login

3. Pemesanan Jasa Layanan



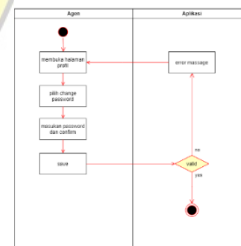
Gambar 9. Usecase Pemesanan Jasa

4. Sign Out



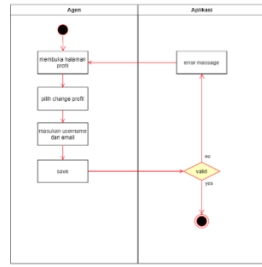
Gambar 10. Usecase Sign Out

5. Change Password



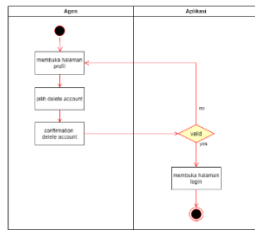
Gambar 11. Usecase Change Password

6. Change Profile



Gambar 12. Usecase Change Profile

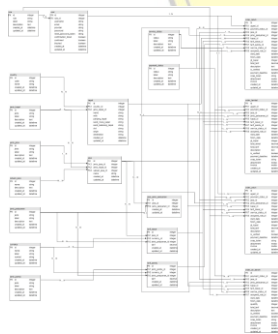
7. Delete Account



Gambar 13. Usecase Delete Account

3. Class Diagram

Diagram kelas adalah jenis diagram struktur UML yang menampilkan struktur, atribut, metode, dan hubungan antar objek, tanpa menjelaskan interaksi di antara kelas-kelas tersebut.



Gambar 14 Class Diagram

4.4 Pembentukan Prototype

Pengerjaan perancangan aplikasi ini dikerjakan menggunakan framework Flutter dengan menggunakan vscode sebagai text editor. Pada pembentukan prototype mengimplementasikan clean arsitektur yang dimana dibagi menjadi 3 bagian yaitu presentation, data dan domain.

1. Presentation Layer

Pada tahap ini presentation layer implementasi tampilan atau user interface kepada pengguna dan interaksi pengguna dengan sistem. BLoC mengelola state dan aliran data sehingga fokus menampilkan UI, memisahkan logika bisnis dari presentasi aplikasi. Folder bloc berisi file seperti cubit atau bloc yang menangani events dan state aplikasi.

2. Data Layer

Tahap ini berfungsi untuk mengelola data dalam aplikasi agar kode lebih terorganisir, mudah dipelihara dan lebih mudah untuk diuji. Dengan memisahkan model, repository dan service memungkinkan aplikasi tetap fleksibel dalam mengelola data dari berbagai sumber.

3. Domain Layer

Pada tahap tersebut berisi logika bisnis dan sebagai jembatan antara presentation layer dan data layer.

Folder usecases berisi logika aplikasi yang berhubungan langsung dengan kebutuhan bisnis atau alur kerja pengguna.

4.5 Deployment Delivery & Feedback

1. Blackbox Testing

Blackbox testing berkonsentrasi pada pengujian sistem dari luar tanpa memperhatikan struktur internalnya.

Tabel 1. Blackbox Testing

Fitur	Test Data	Hasil
Login	Email: amandaputri.ap23@gmail.com Password: manda123	Berhasil 1
Login	Email: amandaputri.ap23@gmail.com	Gagal

	mail.com Password: manda101	
Login	Email: amandaputri.ap23@gmail.com Password: -	Gagal
Login	Email: - Password: manda123	Gagal
Signup	Username: manda Email: amandaputri.ap23@gmail.com Password: manda123	Berhasil
Change profil	Username: khalda Email: amandaputri.ap23@gmail.com	Berhasil
Change password	Current Password: manda123 New Password: amanda123 Password confirm: amanda123	Berhasil
Pemesanan Jasa	Start date: 08/04/2024 21:30 Finish date: 08/12/2024 08:30 quantity: 1000 total_tarif: 200,000 description: tiba tepat waktu	Berhasil
SignOut	Klik button signout	Berhasil
Delete Account	Klik button delete account	Berhasil

2. User Acceptance Testing

Pengujian ini bertujuan untuk mengetahui seberapa sesuai sistem dengan kebutuhan pengguna.

Tabel 2. Kriteria Skor 1

Skala	Keterangan	Skor	Presentase
SS	Sangat Sesuai	5	80 – 100%
S	Sesuai	4	60 - 79%
CS	Cukup Sesuai	3	40 – 59%
TS	Tidak Sesuai	2	20 – 39%

STS	Sangat Tidak Sesuai	1	0 – 19 %
-----	---------------------	---	----------

Tabel 3. Kriteria Skor 2

Skala	Keterangan	Skor	Presentase
SM	Sangat Mudah	5	80 – 100%
M	Mudah	4	60 - 79%
CM	Cukup Mudah	3	40 – 59%
TM	Tidak Mudah	2	20 – 39%
STM	Sangat Tidak Mudah	1	0 – 19 %

Tabel 4. Hasil Angket

Pertanyaan	Frekuensi Jawaban				
	S S/ S M	S / M	C S/ C M	T S/ T M	ST S/S TM
Apakah tata letak antarmuka aplikasi mudah dipahami?	11	1 7	2	-	-
Apakah anda merasa nyaman dengan pilihan warna dan gaya visual pada aplikasi?	12	1 5	3	-	-
Apakah ukuran teks dan ikon pada aplikasi cukup besar untuk dilihat dengan jelas?	10	1 7	3	-	-
Apakah elemen-elemen navigasi (seperti tombol dan menu) mudah ditemukan dan digunakan?	12	1 6	2	-	-
Apakah langkah-langkah untuk memesan layanan (labuh, tambat, yacht, air bersih) jelas dan mudah diikuti?	12	1 6	2	-	-
Apakah anda mengalami	14	1 2	4	-	-

kesulitan saat menggunakan fitur layanan kapal?					
Apakah anda merasa mudah mengubah informasi profil dan kata sandi melalui aplikasi?	15	1 3	2	-	-
Apakah Anda merasa aplikasi ini membantu dalam menyelesaikan proses pemesanan layanan kapal secara efisien?	16	1 2	2	-	-
Total	10 2	1 1 8	20		

Berdasarkan skor yang telah ditetapkan dapat dihitung sebagaimana berikut:
 Jumlah skor SS/SM = $102 \times 5 = 510$
 Jumlah skor S/M = $118 \times 4 = 472$
 Jumlah skor CS/CM = $20 \times 3 = 60$
 Jumlah skor TS/TM = $0 \times 2 = 0$
 Jumlah skor STS/STM = $0 \times 1 = 0$
 Jumlah Total Skor = 1.042

Nilai tertinggi = $30 \times 8 \times 5 = 1.200$.
 $P = \frac{s}{skor} \times 100\%$
 $= \frac{1042}{1200} \times 100\% = 86,83\%$

5. KESIMPULAN

Penelitian ini berhasil menerapkan Clean Architecture pada aplikasi layanan kapal PT ABC berbasis Flutter, dengan pembagian kode menjadi tiga layer: presentation, data, dan domain. Pendekatan ini memudahkan pengelolaan dan pemeliharaan kode, serta meningkatkan efisiensi pengembangan. Pengujian Blackbox menunjukkan bahwa aplikasi berfungsi dengan baik, sementara User Acceptance Testing (UAT) dengan hasil 86,83% memastikan aplikasi memenuhi kebutuhan pengguna dan meningkatkan responsivitas perusahaan

terhadap perubahan pasar. Clean Architecture terbukti efektif dalam mengoptimalkan operasi bisnis PT ABC, memudahkan pemeliharaan, dan meningkatkan pemahaman struktur kode bagi anggota tim baru. Untuk penelitian selanjutnya dapat menambahkan fitur layanan barang di pelabuhan dan notifikasi status, membandingkan Clean Architecture dengan arsitektur lain seperti MVVM untuk mengukur efektivitas dan efisiensi pengembangan.

6. UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada PT. ABC atas dukungan penuh dan kontribusinya dalam penelitian ini. Partisipasi dan kerjasama yang diberikan sangat membantu dalam pengembangan aplikasi layanan kapal berbasis mobile yang inovatif ini.

DAFTAR PUSTAKA

- Adisty, N. (2022). *Mengulik Perkembangan Penggunaan Smartphone di Indonesia*. <https://goodstats.id/article/mengulik-perkembangan-penggunaan-smartphone-di-indonesia-st2LA>
- Aditya, R., Handrianus Pranatawijaya, V., Bagus Adidyana Anugrah Putra, P., Hendrik Timang, J., Palangkaraya, K., & Tengah, K. (2021). Rancang Bangun Aplikasi Monitoring Kegiatan Menggunakan Metode Prototype. In *JOINTECOMS (Journal of Information Technology and Computer Science) p-ISSN: xxxx-xxxx* (Vol. 1, Issue 1).
- Chamida, M. A., Susanto, A., & Latubessy, A. (2021). Analisa User Acceptance Testing Terhadap Sistem Informasi Pengelolaan Bedah Rumah Di Dinas Perumahan Rakyat Dan Kawasan Permukiman Kabupaten Jepara. *Indonesian Journal of Technology, Informatics and Science (IJTIS)*, 3(1), 36–41. <https://doi.org/10.24176/ijtis.v3i1.7531>
- Eka Yuda Wibawa, A. (2021). Implementasi Platform Digital Sebagai Media Pembelajaran Daring Di Mi Muhammadiyah Pk Kartasura Pada

- Masa Pandemi Covid-19. *Berajah Journal*, 1(2), 76–84. <https://doi.org/10.47353/bj.v1i2.15>
- Kinaswara, T. A., Hidayati, N. R., & Nugrahanti, F. (2019). *Rancang Bangun Aplikasi Inventaris Berbasis Website pada Kelurahan Bantengan*.
- Margareta, M. N., & Putra, S. H. (2022). *Sistem Informasi Pengolahan Data Pasien pada Klinik Essiva Berbasis Web dengan Metode Prototype*. 6(April), 330–339. <https://doi.org/http://doi.org/10.33395/remik.v6i2.11562>
- Nelly Sofi, & Riza Dharmawan. (2022). PERANCANGAN APLIKASI BENGKEL CSM BERBASIS ANDROID MENGGUNAKAN FRAMEWORK FLUTTER (BAHASA DART). *Jurnal Teknik Dan Science*, 1, 53–64.
- Putra, A. P., Andriyanto, F., Karisman, Dewi Muji Harti, T. D. M., & Puspitasari, W. (2020). *PENGUJIAN APLIKASI POINT OF SALE BERBASIS WEB MENGGUNAKAN BLACK BOX TESTING*. <https://doi.org/https://doi.org/10.33557/binakomputer.v2i1.757>
- Silvi Purnia, D., Surahman, M., & Agustin, W. (2021). Implementasi Metode Prototyping Pada Rancang Marketplace Rumah Kost Berbasis Mobile. *Jurnal Sains Dan Manajemen*, 9(1).
- Abdillah, F., Sirojudin, A., Amin, M. Y., Setiawan, E., & Atmadji, J. (2024). ANALISIS SKALABILITAS APLIKASI SMART PARKING MELALUI PENERAPAN CLEAN ARCHITECTURE. 8(1), 96–104.
- Alhazmy, N. A., Chandra, Z. N., Atmadiputra, P., & Triyana, Y. (2023). Building a Comprehensive Content Management System with NPM, Vue.js, Node.js, Postgresql, and Strap. In *Liaison Journal Of Engineering* (Vol. 3, Issue 1).
- Asosiasi Penyelenggara Jasa Internet Indonesia. (2022). *APJII di Indonesia Digital Outlook 2022*. https://apjii.or.id/Berita/d/Apjii-Di-Indonesia-Digital-Outlook-2022_857.
- Badrudduja, Moh. H., & Putra, R. E. (2022). Penerapan Clean Architecture pada Aplikasi Pemesanan Makanan menggunakan Metode Slope One Algorithm. *Journal of Informatics and Computer Science (JINACS)*, 3(04), 506–514. <https://doi.org/10.26740/jinacs.v3n04.p506-514>
- Devina Vania Chrestella, Erandaru, & Jacky Cahyadi. (2022). Perancangan Mobile-Apps “Woofland” Platform Adopsi Anjing di Surabaya. *Jurnal DKV Adiwarna, Universitas Kristen Petra*, 1, 121–131.
- Hidayatulloh, S., Radiah, U., & Umma, F. K. (2021). Rancang Bangun Sistem Informasi Shipbroker Sebagai Media Transaksi Sewa Menyewa Kapal Berbasis Web. In *Jurnal* (Vol. 3, Issue 1). <http://ejournal.bsi.ac.id/ejurnal/index.php/infortech85>
- Nelly Sofi, & Riza Dharmawan. (2022). PERANCANGAN APLIKASI BENGKEL CSM BERBASIS ANDROID MENGGUNAKAN FRAMEWORK FLUTTER (BAHASA DART). *Jurnal Teknik Dan Science*, 1, 53–64.
- Prasetya, A. F., Sintia, & Putri, U. L. D. (2022). Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (Unified Modelling Language). In *DOI: ...* (Vol. 1, Issue 1).
- Saifulloh, T., Putra Kharisma, A., & Brata, D. W. (2023). Pengembangan Aplikasi Perangkat Bergerak Pencarian Partner Lomba berbasis Android menggunakan Clean Architecture (Vol. 7, Issue 4). <http://j-ptiik.ub.ac.id>