

## Pembuatan Sistem Informasi Pariwisata Kabupaten Kotawaringin Barat Kalimantan Tengah berbasis Aplikasi *Mobile* dengan Menggunakan Integrasi *REST API*

Joshua Arnold Tradanarova<sup>1</sup>, Tibyani<sup>2</sup>, Widhy Hayuhardika Nugraha Putra<sup>3</sup>

Program Studi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Brawijaya

Email: <sup>1</sup>iamjarn98@student.ub.ac.id, <sup>2</sup>tibyani@ub.ac.id, <sup>3</sup>widhy@ub.ac.id

### Abstrak

Pariwisata termasuk salah satu sektor yang terdampak akibat pandemi *COVID-19* di Indonesia, sehingga terjadi penurunan cukup signifikan dari segi jumlah pengunjung. Hal itulah yang kemudian menjadi pengaruh terhadap pertumbuhan pendapatan negara, devisa, dan lapangan pekerjaan. Sebab di antara bidang pariwisata dan bidang perekonomian, keduanya saling terkait dan terikat. Agar bidang pariwisata dan ekonomi kreatif kembali beranjak, maka dilakukanlah budaya giat dalam pemanfaatan layanan dan teknologi *digital* guna menciptakan interaksi antara masyarakat dan sistem, jadi diharapkan akan menghasilkan sinergi yang luar biasa. Pada pengimplementasian sistem berbasis *web admin*, menggunakan bahasa pemrograman *PHP* di dalam *framework Laravel*. Sedangkan pengimplementasian sistem berbasis aplikasi *mobile* menggunakan bahasa pemrograman *Dart* di dalam *framework Flutter*, dengan mendukung integrasi dari *REST API* terkait pertukaran informasi antara data-data dari *database admin* dengan aplikasi *mobile* yang bertajuk *KOBAR Connect*. Dari hasil pengujian yang dijalankan, pada metode *Black Box Testing*, menghasilkan 17 buah kebutuhan fungsional berhasil diuji aksesibilitasnya, sedangkan 1 buah kebutuhan gagal diuji. Sedangkan pada metode *System Usability Scale (SUS)* yang melibatkan 58 responden, Penulis memperoleh skor *SUS* rata-rata bernilai 68,88. Sehingga aplikasi ini memperoleh predikat D secara skala kualitas, predikat *OK* yang hampir mendekati *Good* secara *adjective rating*, dan status *Marginal – High* dalam tingkat penerimaan. Artinya, aplikasi ini membutuhkan perbaikan sistem lagi ke depannya agar tingkat penerimaannya lebih baik.

**Kata kunci:** *Flutter, Kotawaringin Barat, Laravel, Pariwisata, REST API*

### Abstract

*Tourism is one of the sectors that has been affected by the COVID-19 pandemic in Indonesia, resulting in a significant decrease in the number of tourists. It has affected state revenues growth, foreign exchange, and employment. There are interrelated and bound between the tourism and the economic sector. In order for the tourism and the creative economy to move forward on, an enterprising culture of utilizing digital services and technology is being carried out to create interaction between the community and the system, so it is hoped that it will produce extraordinary synergies. The implementation of web admin-based system is done using the PHP language within the Laravel framework. But the implementation of mobile application-based system is carried out using the Dart language within the Flutter framework, by supporting the REST API's integration related to the information exchange between data from the admin database and mobile application called KOBAR Connect. From the results of the testing, 17 functional requirements were successfully tested for accessibility, while 1 requirement failed to be tested when using the Black Box Testing. While at the System Usability Scale (SUS) method which it was involved 58 people, the author obtained an average SUS score of 68.88. So, this application received the D predicate in grade scales, OK almost close to Good in adjective rating, and the Marginal – High in acceptance status. It means that this application requires more system improvements in the future so that at least to get better acceptance range.*

**Keywords:** *Flutter, Kotawaringin Barat, Laravel, REST API, Tourism*

## 1. PENDAHULUAN

Pada awal pandemi menyerang Indonesia, Pemerintah Kotawaringin Barat memutuskan menutup semua destinasi wisata di wilayah setempat, seiring meningkatnya wabah *COVID* ditambah kurangnya promosi dari Pemerintah setempat, kunjungan wisata ke wilayah Kotawaringin Barat semakin anjlok. Hingga pada September 2021 lalu, secara perlahan Pemerintah Kotawaringin Barat memutuskan membuka kembali destinasi wisata namun hanya berdasarkan zona hijau atau kuning yang ditetapkan oleh Tim Satgas *COVID-19* setempat, bahwa jumlah pengunjung dibatasi menjadi 25 persen, dan waktu operasional tempat wisata atau hiburan berlangsung hingga pukul 20:00 WIB dengan persyaratan menjalankan protokol kesehatan.

Pemerintah Kotawaringin Barat harus terus menggali potensi-potensi di sekitar bidang pariwisata, serta mengembangkan inovasi dalam strategi pariwisata di tengah pandemi masih melanda, baik itu diterjemahkan ke dalam bentuk layanan, *object positioning*, serta pemanfaatan teknologi digital guna menciptakan pengalaman batasan dalam berinteraksi dengan orang lain, agar bidang pariwisata serta ekonomi kreatif tetap bertahan di tengah pandemi, sampai semuanya menjadi normal seperti semula.

*REST* memiliki *resources* yang dimanipulasi dengan perpindahan representasi data melalui alamat *HTTP* yang unik dengan metode kerja *PUT*, *GET*, *POST*, dan *DELETE*. Dan setiap representasinya harus mengimplementasikan *CRUD* untuk mendukung arsitektur *REST* itu sendiri (Sinha, Khatkar, & Gupta, 2014). *REST API* tidak boleh bergantung pada protokol komunikasi tunggal. Secara umum, elemen protokol apapun yang menggunakan *URI* untuk hal identifikasi, harus mengikuti skema *URI* yang digunakan. (Fielding, 2008).

Alasan Penulis melakukan penelitian dengan mengembangkan aplikasi *mobile* untuk pengguna adalah karena ponsel atau *smartphone* memudahkan pengguna untuk melakukan berbagai aktivitas sesuai kebutuhan mereka, sehingga diharapkan lebih menjangkau daya minat pengguna terhadap aplikasi itu sendiri (Wicaksono, 2022).

Dan alasan Penulis melakukan penelitian dengan membutuhkan integrasi *REST API* adalah *REST* menggunakan format *JSON*, yang

memungkinkan pertukaran data yang efisien dan terstruktur antara data-data yang tersimpan dalam sistem berbasis *web admin* dengan aplikasi *mobile* untuk pengguna, sehingga aplikasi *mobile* secara otomatis diperbarui tanpa harus menyegarkan halaman aplikasi (3 Alasan Kenapa Harus Menggunakan *REST API Protocol*, 2022).

## 2. LANDASAN PUSTAKA

### 2.1. Kajian Penelitian

Berikut adalah beberapa kajian penelitian yang digunakan sebagai acuan dalam melakukan penelitian dan pengembangan sistem informasi pariwisata dan pengimplementasian *REST API*.

Acuan pertama adalah penelitian yang dilakukan oleh Arsad (2011), yaitu mengembangkan sistem informasi pariwisata berbasis *web* di Dinas Kebudayaan dan Pariwisata Kabupaten Muna. Pengujung bisa mengakses profil, kalender, agenda kegiatan, serta galeri tentang kesenian tradisional, wisata alam, kuliner, serta informasi hotel atau kabupaten di Kabupaten Muna. Sedangkan admin diberikan akses untuk melakukan pembaruan konten (*CRUD*) terhadap agenda kegiatan, galeri, informasi kegiatan, dsb. Pengimplementasian sistem dilakukan dengan *framework PHP* dan *database MySQL*.

Acuan berikutnya, dilakukan oleh Baharuddin, Hamka Wakkang, dan Bambang Irianto (2022) melakukan implementasi *Web Service* terhadap data *COVID-19* di Sulawesi Selatan, menggunakan integrasi *REST API*, untuk mengetahui perkembangan kasus *COVID-19* di setiap kota dan kabupaten dalam wilayah Sulawesi Selatan. Sudut pandang yang diambil dari artikel ini adalah teknologi *web service* dengan integrasi *REST API*. Hasil implementasi tersebut dapat diterapkan di perangkat berbasis *web* dan *mobile* sebagai *requestor service*, sehingga pengambilan data antara sistem data *JSON* yang dibuat menggunakan *API* dapat terbentuk dan integrasi data dapat dilakukan secara terpusat.

### 2.2. Sistem Informasi

Sistem informasi adalah komponen yang berperan dalam mengumpulkan, memproses, menyimpan serta mendistribusikan data dan informasi untuk mendukung pengambilan keputusan dan pengendalian dalam suatu organisasi. (Laudon & Laudon, 2014).

### 2.3. Kabupaten Kotawaringin Barat

Kabupaten Kotawaringin Barat resmi didirikan pada tanggal 3 Oktober 1959. Wilayah ini memiliki visi *Gerakan Membangun Kotawaringin Barat Menuju Kejayaan dengan Kerja Nyata dan Ikhlas*, karena itulah semboyan dari Kotawaringin Barat adalah *Marunting Batu Aji* (Menuju Kejayaan). Serta memiliki beberapa misi untuk memperkuat tata pemerintahan yang bersih, demokratis, dan transparan, meningkatkan kualitas hidup manusia, kemandirian ekonomi berbasis pertanian, kelautan, industri serta potensi daerah, serta pelestarian kesenian dan kebudayaan lokal. (Gambaran Umum, n.d.)

### 2.4. Laravel

*Laravel* adalah *framework* yang membantu mengoptimalkan penggunaan *PHP* dengan konsep *MVC* (*Model, View, Controller*) dalam pengembangannya, sehingga *web* yang dibangun menjadi lebih cepat, aman, dan praktis tanpa meningkatkan biaya pengembangan awal serta pemeliharaan perangkat lunak. (History Of Laravel, 2021)

### 2.5. Flutter

*Flutter* merupakan teknologi *framework* secara *open source* (bebas dan terbuka), yang dikembangkan oleh *Google* dan resmi dirilis pada bulan Mei 2017 lalu. Konsep dari *Flutter* itu sendiri adalah *framework* pendukung aplikasi dengan tampilan antarmuka yang menarik, serta dapat dijalankan secara *native* ke dalam aplikasi berbasis *web*, *mobile*, dan *desktop* hanya dari satu basis kode. *Flutter* sendiri menggunakan bahasa pemrograman *Dart*, bahasa yang dikembangkan sendiri oleh *Google*. (Sachindana, 2021).

### 2.6. REST API

*REST API* (*Representational State Transfer*) adalah bentuk penerapan dari *Application Programming Interface*, yang di dalamnya mendefinisikan beberapa aturan yang membatasi para *programmer* untuk mengakses suatu *database*. Jika *API* berperan sebagai penghubung, maka *REST* yang berlaku sebagai pedoman *programmer* dalam merancang *web service*. Dalam *REST API*, kita mengenal istilah *request* dan *response* pada data yang akan dikirimkan atau akan diterima. Data yang

dikirimkan dalam *REST API* umumnya berformat *JSON*.

Dalam cara kerjanya, hasil *request* data dari *user* ketika dikirimkan akan berubah formatnya dari *JSON* menjadi *REST Server*. Dan *REST Server* menerima *JSON response* beserta kode status dan mengirimkannya ke *server*, lalu *server* menerima *response* dari *REST Server* dan mengubah *response* menjadi *HTML* (Fielding, 2008).

### 2.7. SDLC – Waterfall

*SDLC* (*Software Development Life Cycle*) adalah representasi berupa kerangka dari proses pengembangan perangkat lunak, dimana masing-masing representasi model menunjukkan proses-proses terstruktur berdasarkan perspektif model yang digunakan dalam pengembangan sistem.

*Waterfall* adalah salah satu model pengembangan perangkat lunak dengan konsep sistematis dan berurutan, yang masing-masing prosesnya direpresentasikan secara terpisah, mulai dari spesifikasi kebutuhan, perancangan desain, pengimplementasian, pengujian hingga pemeliharaan. (Sommerville, 2011)

### 2.8. Unified Modelling Language

*Unified Modelling Language* adalah alat yang berperan dalam memvisualisasikan hasil analisa dan desain serta menggambarkan pemodelan sebuah sistem dengan pendekatan berorientasi obyek (Braun, Sivilis, Shapiro, & Verstegh, 2001)

UML terbagi menjadi empat jenis diagram, yakni *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.

#### 1. Use Case Diagram

*Use Case Diagram* adalah diagram yang memperlihatkan himpunan berupa kebutuhan sistem dengan aktor yang menjalankan sistem tersebut untuk mendefinisikan fitur yang harus disediakan oleh sistem dan menyatakan sifat sistem dari sudut pandang pengguna (Murad, Kusniawati, & Asyanto, 2013).

#### 2. Activity Diagram

Adalah urutan aktivitas yang dirangkai untuk memaparkan secara visual alur data pada *use case diagram*. *Activity diagram* menjelaskan jalannya sebuah aktivitas yang dimulai dengan simbol *initial node* sampai pada akhir yang digambarkan dengan simbol *activity final* (Waykar, 2013).

3. Sequence Diagram

Sequence diagram menggambarkan aliran pesan yang terjadi antarkelas yang dideskripsikan pada class diagram dengan merujuk pada alur sistem activity diagram yang telah dibuat sebelumnya (Vidia, 2013).

4. Class Diagram

Class diagram adalah jenis diagram dalam UML yang menunjukkan sistem kelas, atribut, metode, dan hubungan antarobjek sebagai analisis bisnis dalam pembuatan model sistem. (What is Class Diagram?, n.d.)

2.8. Black Box Testing

Black Box Testing adalah metode pengujian perangkat lunak yang dilakukan pengguna guna memastikan apakah sistem menghasilkan keluaran yang sesuai dengan instruksi sistem yang diharapkan. Metode Black Box menjadi efektif bilamana langsung dilakukan setelah pengembangan perangkat lunak, terlebih perangkat lunak yang dibangun tersebut bersifat besar dan kompleks (Black Box Testing, n.d.).

2.9. System Usability Scale

System Usability Scale adalah metode pendekatan pengujian berbentuk kuesioner, untuk mengetahui seberapa besar tingkat usability sebuah produk atau sistem, sehingga melibatkan sekumpulan responden sebagai penguji jalannya produk atau sistem tersebut.

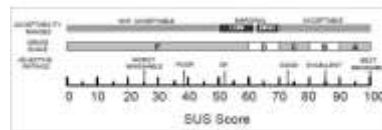
Metode ini diciptakan oleh John Brooke pada tahun 1986. Bahkan hingga saat ini metode ini menjadi acuan dalam menguji usability suatu produk, karena mudah diterima oleh responden serta terbukti valid dalam menentukan kelayakan sistem.

Dalam konsep SUS, terdapat 10 butir pertanyaan, dimana pertanyaan bernomor ganjil bermuatan positif, sedangkan pertanyaan bernomor genap bermuatan negatif, dan model skala penilaiannya pun menggunakan skala Likert, yang memiliki rentang poin satu hingga lima.

Cara menghitung hasil pengukuran system usability scale adalah :

1. Kurangi skor asli setiap pertanyaan bernomor ganjil dengan satu (x - 1).
2. Kurangi skor asli setiap pertanyaan bernomor genap dari 5 (5 - x).
3. Akumulasikan skor SUS dari pertanyaan

bernomor ganjil dan genap, lalu kalikan dengan 2,5 (Will, 2017)



Gambar 2.1 : Skala Penilaian System Usability Scale (Bangor, Kortum, & Miller, 2009)

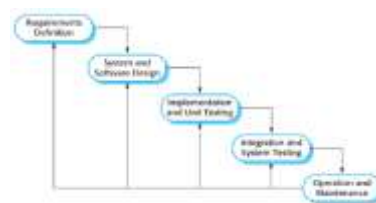
3. METODOLOGI PENELITIAN

Bagian ini akan memaparkan langkah-langkah melakukan penelitian, mulai dari observasi, penentuan, perancangan antarmuka, implementasi, pengujian sistem sampai pada bagian kesimpulan dan saran. Gambar 3.1. akan memaparkan tahapan dalam penelitian ini.



Gambar 3.1 : Tahapan Metodologi Penelitian

Untuk penelitian ini, Penulis menerapkan model pengembangan perangkat lunak bertajuk Waterfall, pengembangan dengan tahapan mengalir ke bawah seperti air terjun, karena menggambarkan tahapan yang berurutan dan sistematis, mulai dari tahapan analisis kebutuhan, perancangan desain, pengimplementasian sistem, pengujian sistem, hingga pemeliharaan perangkat lunak.



Gambar 3.2 : Tahapan dalam Model Pengembangan Waterfall (Sommerville, 2011)

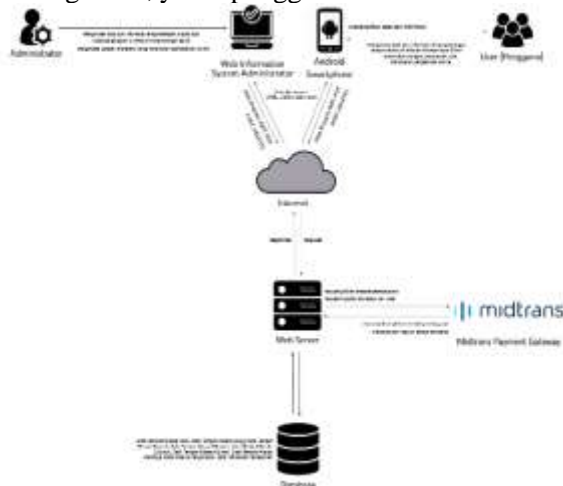


**4. ANALISIS KEBUTUHAN SISTEM**

Berikut adalah salah satu gambar dari masing-masing pemodelan sistem, baik dari *use case diagram*, *activity diagram*, *sequence diagram*, serta *class diagram*.

**4.1. Landasan Model Sistem**

Alur model ini dibuat sebagai landasan dalam mengembangkan sistem dengan memaparkan peranan-peranan pada masing-masing aktor, yakni pengguna dan administrator.



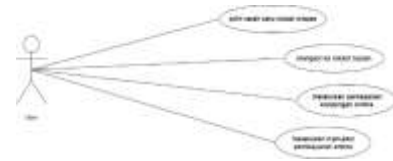
**Gambar 4.1 : Landasan Model Sistem Informasi Pariwisata Kabupaten Kotawaringin Barat**

**4.2. Use Case Diagram**

Mengidentifikasi peran aktor dalam menjalankan fitur apa saja dalam sistem yang dibangun.



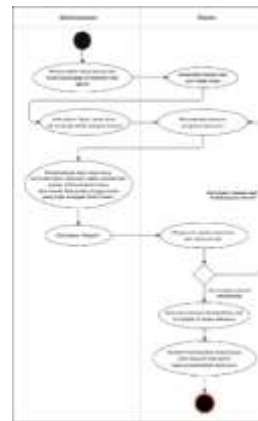
**Gambar 4.2 : Use Case Diagram Peran Administrator terhadap Sistem**



**Gambar 4.3 : Use Case Diagram Peran Pengguna terhadap Sistem**

**4.3. Activity Diagram**

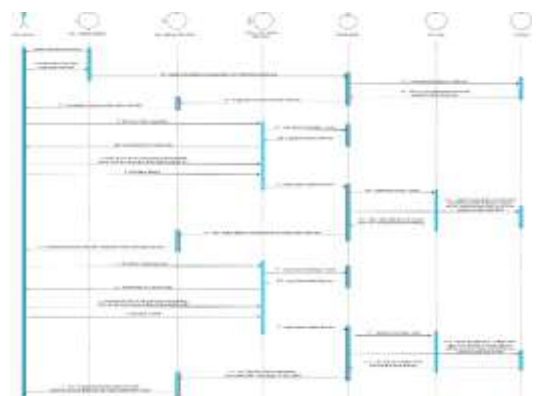
Memaparkan aktivitas sebagai bentuk aksi dan reaksi dari aktor dan sistem ketika menjalankan salah satu fitur.



**Gambar 4.4 : Activity Diagram – Admin Tambah Data Lokasi**

**4.4. Sequence Diagram**

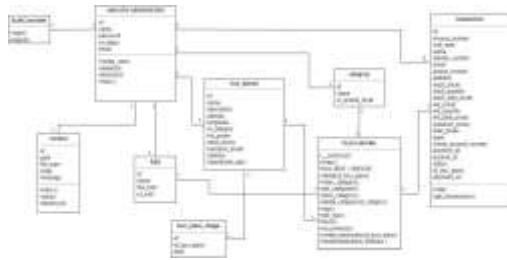
Memaparkan urutan proses jalannya kegiatan menjalankan salah satu fitur dari segi aktor serta sistem yang terbagi atas *view page*, *controller*, dan *model (database)*.



**Gambar 4.5 Sequence Diagram – Admin Tambah Data Lokasi**

**4.5. Class Diagram**

Berikut adalah perancangan *class diagram* untuk menggambarkan pemodelan sistem informasi pariwisata Kotawaringin Barat.



Gambar 4.6 : Class Diagram

4.6. Physical Data Model

Berikut adalah perancangan basis data untuk menggambarkan pemodelan sistem informasi pariwisata Kotawaringin Barat.



Gambar 4.7 : Physical Data Model

4.7. Spesifikasi Perangkat

Dalam membangun sebuah sistem, diperlukan adanya spesifikasi perangkat keras untuk mengetahui perangkat apa saja yang dibutuhkan dalam proses perancangannya. Berikut adalah tabel spesifikasi perangkat keras untuk perancangan sistem ini.

Tabel 4.1 : Spesifikasi Perangkat Keras dalam Pengembangan Sistem

Komponen	Spesifikasi
System Model	Lenovo IdeaPad 3-14ALC6 Laptop - Type 82KT 82KT00V8US PF3QLJLV
Processor	AMD Ryzen 7 5700U with Radeon Graphics, 1801 Mhz, 8 Core(s), 16 Logical Processor(s)
Memory (RAM)	8192MB RAM
Graphic Card	AMD Radeon Graphics
Operating System	Microsoft Windows 11 Home Version 10.0.22621 Build 22621

Tabel 4.2 : Spesifikasi Perangkat Keras dalam Mengakses Aplikasi Mobile

Komponen	Spesifikasi
Processor Smartphone Minimum	1,2 GHz
Memory (RAM)	4 GB

ROM	16 GB
OS Minimum	Android 9.0 (Pie)
Processor Smartphone Minimum	1,2 GHz
Memory (RAM)	4 GB
ROM	16 GB
OS Minimum	Android 9.0 (Pie)

Selain spesifikasi perangkat keras, diperlukan juga adanya spesifikasi perangkat lunak dalam merancang sistem ini. Berikut adalah tabel spesifikasi perangkat lunak untuk penggunaan sistem ini :

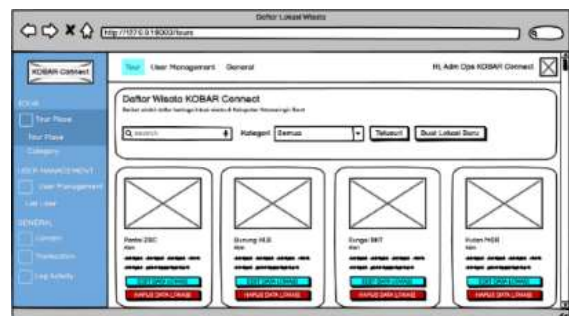
Tabel 4.3 : Spesifikasi Perangkat Lunak dalam Pengembangan Sistem

Komponen	Spesifikasi
Development Tool	Visual Studio Code (perancangan web admin dan aplikasi mobile)
Server Tool	Apache (web server untuk jaringan lokal)  Ngrok (proxy tunnel untuk menghubungkan jaringan lokal ke ponsel)
Database Tool	MySQL
Framework	PHP versi 8.2.0, Laravel versi 8.83.27 (pengembangan sistem berbasis web admin), Flutter (pengembangan sistem berbasis aplikasi mobile)
Web Browser	Google Chrome & Microsoft Edge (akses halaman web admin)

5. PERANCANGAN & IMPLEMENTASI SISTEM

5.1. Perancangan Antarmuka

Gambar di bawah ini adalah perancangan antarmuka dari halaman daftar lokasi wisata pada web admin, yang dirancang melalui program Balsamiq Wireframes.



Gambar 5.1 : Perancangan Antarmuka Halaman Daftar Lokasi Wisata pada Web Admin

Gambar berikutnya adalah perancangan antarmuka dari fitur melakukan pemesanan kunjungan secara *online* pada aplikasi *mobile*, yang dirancang melalui program *Adobe XD*.



Gambar 5.2 : Perancangan Antarmuka Halaman Pemesanan *Online* pada Aplikasi *Mobile* (1)



Gambar 5.3 : Perancangan Antarmuka Halaman Pemesanan *Online* pada Aplikasi *Mobile* (2)



Gambar 5.4 : Perancangan Antarmuka Halaman Konfirmasi Pemesanan pada Aplikasi *Mobile*

### 5.2. Implementasi Sistem

Sedangkan pada tahap pengimplementasian sistem, implementasi untuk web admin menggunakan bahasa pemrograman *PHP* serta *framework Laravel*, dan implementasi untuk aplikasi *mobile* menggunakan bahasa pemrograman *Dart* serta *framework Flutter*.

Gambar di bawah ini adalah implementasi halaman daftar lokasi wisata pada *web admin*.



Gambar 5.5 : Implementasi halaman daftar lokasi wisata pada *web admin*

Gambar di bawah ini adalah implementasi fitur konfirmasi terhadap pemesanan kunjungan secara *online* hingga pada pemilihan fitur pembayaran pada aplikasi *mobile*.



Gambar 5.6 : Implementasi Halaman Konfirmasi Melakukan Pemesanan *Online* pada Aplikasi *Mobile*



Gambar 5.7 : Implementasi Halaman Pemilihan Metode Pembayaran di Aplikasi *Mobile*

## 6. PENGUJIAN

### 6.1. Black Box Testing

Telah dilakukan pengujian *Black Box* terhadap 18 kebutuhan fungsional yang masing-masing telah dijabarkan alur pengujiannya. Namun, sayangnya hanya 1 kebutuhan fungsional yang memiliki hasil uji yang gagal, yakni melakukan navigasi ke lokasi tujuan. Kebutuhan fungsional lainnya memiliki hasil uji yang baik, sehingga menghasilkan pengujian dengan *Black Box* terhadap sistem bernilai 94% valid.

## 6.2. System Usability Scale (SUS)

Pengujian *System Usability Scale* terhadap aplikasi *KOBAR Connect* melibatkan 58 responden. Berdasarkan hasil penghitungan skor terhadap 58 responden setelah menggunakan rumus berdasarkan konsep *System Usability Scale*, maka total nilai (jumlah skor *SUS* untuk 1 responden  $\times 2,5$ ) seluruh responden adalah 3.995 poin. Sehingga, dilakukan penghitungan *SUS* rata-rata dengan membagi total nilai dengan banyaknya responden.

$$\text{Hasil akhir SUS} = \frac{\sum xi}{n} = \frac{3.995}{58} = 68,88$$

## 7. KESIMPULAN & SARAN

### 7.1. Kesimpulan

Berdasarkan penelitian yang telah Penulis lakukan dalam membuat sistem informasi pariwisata Kotawaringin Barat yang bertajuk *KOBAR Connect*, dapat Penulis simpulkan sebagai berikut :

1. Pada bagian analisa kebutuhan desain, telah dihasilkan 14 kebutuhan fungsional dan 4 kebutuhan non-fungsional dari sistem berbasis *web admin*, serta 4 kebutuhan fungsional dan 4 kebutuhan non-fungsional dari sistem berbasis aplikasi *mobile*.
2. Perancangan pada *web admin* menghasilkan 17 rancangan antarmuka, sedangkan pada aplikasi *mobile* menghasilkan 8 rancangan antarmuka, serta 8 buah hasil implementasi tabel dalam *database laravel*.
3. *KOBAR Connect* hanya bisa dijalankan jika pengguna menggunakan perangkat seluler berbasis *Android*, dengan minimal spesifikasi *Android* versi 9 (*Pie*).
4. Pada pengujian menggunakan **Black Box Testing**, hanya 1 dari 18 fitur fungsional yang gagal diuji, yakni navigasi pengguna ke lokasi tujuan dengan integrasi *Google Maps*. Kegagalan ini disebabkan karena *API Key* yang digunakan hanya untuk memunculkan lokasi berdasarkan titik koordinat semata, bukan untuk melakukan navigasi secara *live-location*. Sehingga hasil pengujian sistem melalui *Black Box Testing* adalah **94% valid**.

5. Pada pengujian menggunakan *System Usability Scale*, diketahui bahwa hasil akhir pengujian yang melibatkan sebanyak **58 responden** bernilai **68.88**, sehingga menjadi aplikasi mobile *KOBAR Connect* ini secara *adjective rating* tergolong ke dalam predikat *OK* yang hampir mendekati *Good*, secara skala kualitas (*grade scale*) memperoleh predikat D, dan secara jangkauan penerimaan (*acceptability ranges*) memperoleh predikat *Marginal – High*. Artinya, tingkat *usability* pada aplikasi ini terbilang cukup aman, namun butuh perbaikan lagi ke depannya untuk meningkatkan mutu penerimaan aplikasi, setidaknya bisa memperoleh predikat *Acceptable*.

### 7.2. Saran

Berdasarkan hasil penelitian mengenai pembuatan sistem informasi ini, saran-saran yang menjadi pertimbangan bagi Penulis untuk meningkatkan sistem ini adalah perlu dibuat fitur registrasi dan *login* bagi pengguna aplikasi *mobile*, adanya ulasan pengguna di halaman informasi lokasi, kejelian dari para administrator dalam merangkai kalimat deskripsi informasi lokasi menjadi menarik guna menjangkau daya minat terhadap aplikasi *KOBAR Connect* sekaligus menjangkau daya minat untuk berwisata ke wilayah Kotawaringin Barat, serta dilakukan pengembangan sistem *KOBAR Connect* menjadi *web-based software* ke depannya. agar bisa diakses di berbagai *web browser* baik di perangkat komputer ataupun ponsel.

## 8. DAFTAR PUSTAKA

- 3 Alasan Kenapa Harus Menggunakan REST API Protocol. (2022, July 27). Retrieved from M Target: <https://mtarget.co/blog/rest-api-protocol/#:~:text=Dengan%20REST%20API%2C%20aplikasi%20dapat,perlu%20me%2Drefresh%20seluruh%20halaman.>
- Arsad. (2011). *Perancangan sistem informasi pariwisata berbasis web di dinas kebudayaan dan pariwisata Kabupaten Muna*. Jakarta: UIN Syarif Hidayatullah Jakarta: Fakultas Sains dan Teknologi.
- Baharuddin, Wakkang, H., & Irianto, B. (2022). Implementasi Web Service Dengan Metode REST API Untuk Integrasi Data



- COVID-19 Di Sulawesi Selatan. *Jurnal Sintaks Logika Vol. 2 No. 1 2022*, 236-241.
- Bangor, A., Kortum, P., & Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4(3), 114-123.
- Black Box Testing*. (n.d.). Retrieved from Revou: <https://revou.co/kosakata/black-box-testing>
- Braun, D., Sivilis, J., Shapiro, A., & Verstegh, J. (2001). Object Oriented Analysis and Design Team. *Kennesaw State University CSIS 4650 - Spring 2001*.
- Fielding, R. (2008, October 20). *REST APIs must be hypertext-driven*. Retrieved from Roy Fielding - Untangled: <https://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>
- Gambaran Umum*. (n.d.). Retrieved from Pemerintah Kabupaten Kotawaringin Barat: <https://portal.kotawaringinbaratkab.go.id/id/gambaran-umum>
- History Of Laravel*. (2021). Retrieved from Java T Point: <https://www.javatpoint.com/history-of-laravel>
- Hosch, W. L., Das, D., Young, G., Gregersen, E., Gaur, A., & Rodriguez, E. (2008, December 30). *Android Operating System*. Retrieved from Britannica: <https://www.britannica.com/technology/Android-operating-system>
- Laudon, K. C., & Laudon, J. P. (2014). *Management Information Systems : Managing The Digital Firm - 13th Global Edition*. Harlow: Pearson Education Limited.
- Murad, D., Kusniawati, N., & Asyanto, A. (2013). Aplikasi Intelligence Website Untuk Penunjang Laporan Paud pada HIMPAUDI Kota Tangerang. *CCIT (Creative Communication and Innovative Technology) Journal*, vol. 7, no. 1, September 2013, 44-58.
- Sachindana, S. (2021, June 22). *Introduction To Flutter*. Retrieved from DEV Community: <https://dev.to/sudarasach/intro-to-flutter-2odk>
- Sinha, R., Khatkar, M., & Gupta, S. C. (2014). Design & Development of a REST based Web service platform for mobile applications integration on Cloud. *IJISSET - International Journal of Innovative Science, Engineering & Technology*, Vol. 1 Issue 7, September 2014, 241-246.
- Sommerville, I. (2011). *Software Engineering 9th Edition*. Boston: Pearson Education (on behalf of Addison-Wesley).
- Waykar, d. Y. (2013). A Study of Importance of UML diagrams: With Special Reference to Very Large-sized Projects. *International Conference on Reinventing Thinking beyond boundaries to Excel*. Faridabad: IMT Faridabad.
- What is Class Diagram?* (n.d.). Retrieved from Visual Paradigm: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>
- Wicaksono, P. (2022, January 9). *5 Alasan Bisnis Perlu Memiliki Mobile Apps*. Retrieved from QuBisa: <https://www.qubisa.com/article/alasan-bisnis-perlu-memiliki-mobile-apps>
- Will. (2017, May 31). *Measuring and Interpreting System Usability Scale (SUS)*. Retrieved from UIUX Trend: <https://uiuxtrend.com/measuring-system-usability-scale-sus/>