

# SKRIPSI

## PENGEMBANGAN SAFIR (SISTEM INFORMASI MANAJEMEN TRAVEL HAJI DAN UMROH) DENGAN MENGUNAKAN METODOLOGI *EXTREME PROGRAMMING*



Disusun Oleh:

N a m a : Fardika Rais Hidayatullah

NIM : 17523192

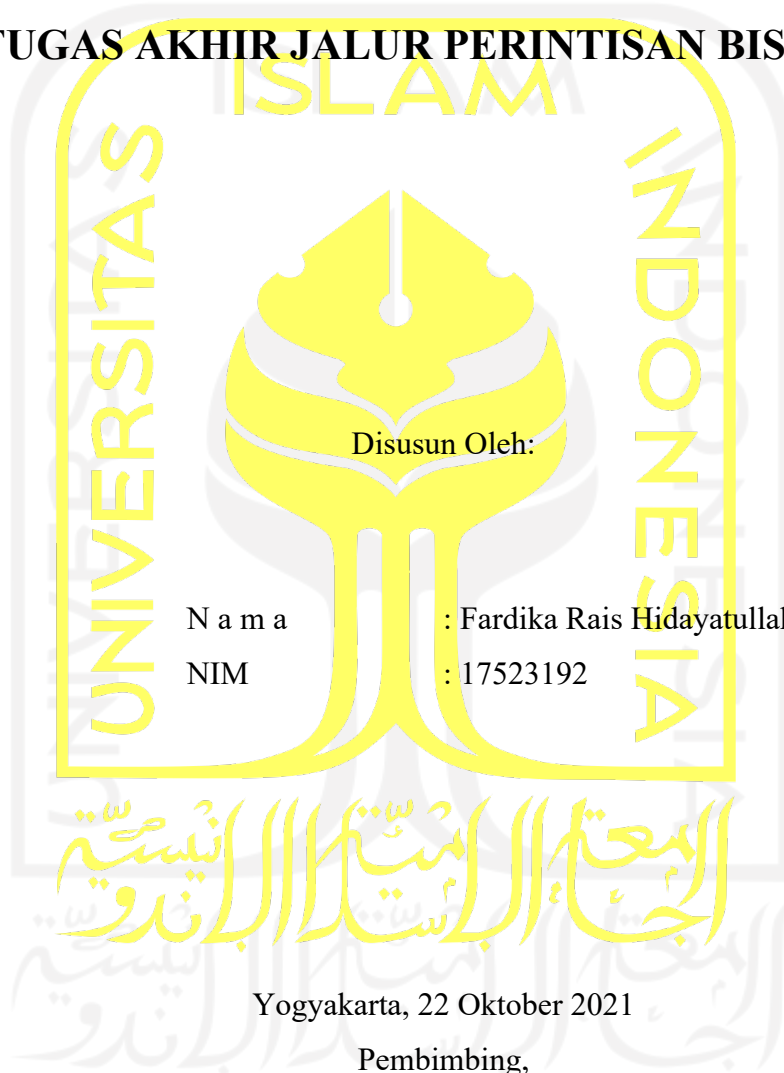
PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA

2021

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN SAFIR (SISTEM INFORMASI  
MANAJEMEN TRAVEL HAJI DAN UMROH) DENGAN  
MENGUNAKAN METODOLOGI *EXTREME PROGRAMMING***

**TUGAS AKHIR JALUR PERINTISAN BISNIS**



( Beni Suranto, S.T., M.SoftEng )

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN SAFIR (SISTEM INFORMASI  
MANAJEMEN TRAVEL HAJI DAN UMROH) DENGAN  
MENGUNAKAN METODOLOGI *EXTREME PROGRAMMING***

**TUGAS AKHIR JALUR PERINTISAN BISNIS**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 23 November 2021

Tim Penguji

**Ketua Penguji**

Beni Suranto, S.T., M.SoftEng.

**Anggota 1**

Hari Setiaji, S.Kom., M.Eng.

**Anggota 2**

Andhik Budi Cahyono, S.T., M.T.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

## HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Fardika Rais Hidayatullah  
NIM : 17523192

Tugas akhir dengan judul:

### **PENGEMBANGAN SISTEM INFORMASI MANAJEMEN TRAVEL HAJI DAN UMROH BERBASIS WEBSITE MENGUNAKAN *LARAVEL FRAMEWORK* DENGAN METODOLOGI *EXTREME PROGRAMMING (XP)* UNTUK SAFIR**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 23 Oktober 2021



(Fardika Rais Hidayatullah)

## HALAMAN PERSEMBAHAN

Semua yang telah berhasil tercapai sampai titik ini merupakan hasil dari dukungan, semangat, motivasi, dan doa yang diberikan dari berbagai pihak. Halaman ini sebagai bagian dari tugas akhir studi Program Studi Informatika Sarjana Universitas Islam Indonesia.

Terimakasih kepada Allah SWT yang tidak hentinya memberikan nikmat yang berlimpah ruah, nikmat sehat, iman, dan ihsan atas ridho yang diberikannya sehingga tugas akhir ini dapat diselesaikan. Segala hal yang kurang berkenan, sekiranya dapat menjadi koreksi bagi pengembangan diri saya sendiri. Bila ada kebermanfaatannya di dalamnya saya bersyukur atas ridho Allah pesan itu tersampaikan.

Tugas akhir ini saya persembahkan untuk semua pihak yang membentuk pribadi saya dan kematangan pola pikir saya. Membantu disaat masa sulit, menemani dikala fikiran tidak berjalan dengan baik, mencaci dan memuji disaat aku mampu menganggungnya. Saya harap kalian menerima dan mengamini. Aamiin.

Yang terhormat ayah saya Ghofarudin yang selalu menjadi acuan saya atas semangatnya sekuat tenaga menjadi tulang punggung bagi keluarga. Tidak pernah lelah dan selalu berusaha atas semua yang saya miliki hingga saat ini, terutama dalam Pendidikan saya selama ini. Ibu saya Ika Kurnia Sari yang selalu saya sayangi, saya rindu, menjadi tempat buat saya dikala saya sulit, sehatlah selalu ibu. Aku bersyukur bisa terlahir karenamu ibu. Semoga Allah senantiasa membalas pengorbanan kalian melalui apapun bentuknya, termasuk saya anakmu. Aamiin.

Dosen dan semua guru yang telah mendidik saya dengan baik melalui Pendidikan formal maupun non formal. Terkhusus untuk pembimbing tugas akhir saya Pak Beni Suranto atas bimbingan untuk menyempurnakan tugas akhir ini. Tidak luput untuk Pak Raden Teduh Dirgahayu dan jajarannya yang membantu saya dan memberi dukungan selama studi Sarjana saya.

## HALAMAN MOTO

Pelajarilah ilmu sebanyak-banyaknya bukan hanya untuk dirimu seorang. Tetapi, jadikanlah ilmu yang kamu pelajari untuk dapat bermanfaat bagi banyak orang dan mengalir seperti sungai tanpa akhiran.

(Aftah Fatahillah)

*Intelligence plus character – that is the goal of true education.*

(Martin Luther King Jr)

*“Allah tidak membebani seseorang melainkan sesuai kesanggupannya”*

*(QS An Najm: 39).*



## KATA PENGANTAR



### **Assalamualaikum Warahmatullahi Wabarakatuh**

Alhamdulillah, segala puji dan syukur saya panjatkan atas kehadiran Allah SWT, yang telah memberikan rahmat dan ridhonya sehingga penulis dapat menyelesaikan laporan tugas akhir ini yang berjudul “Pengembangan Sistem Informasi Manajemen Travel Haji dan Umroh Berbasis Website Menggunakan *Laravel Framework* dengan Metodologi *Extreme Programming (XP)* untuk Safir”. Laporan ini disusun untuk memenuhi Tugas Akhir sebagai syarat untuk menyelesaikan pendidikan pada jenjang strata 1 (S1) pada Jurusan Informatika Universitas Islam Indonesia. Penulis sadar laporan ini tidak akan dapat selesai dengan waktu yang cepat tanpa dukungan serta motivasi dari beberapa pihak. Oleh karena itu penulis tidak lupa menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT, atas limpahan rahmat dan hidayah-Nya yang selalu ada di setiap langkah dalam memberikan kekuatan, kemampuan, dan menjaga semangat saya untuk dapat menyelesaikan Tugas Akhir ini dengan lancar.
2. Kedua orang tua saya untuk doa dan dukungan yang diberikan kepada saya.
3. Bapak Beni Suranto, S.T., M.SoftEng, selaku dosen pembimbing tugas akhir yang memberikan peran luar biasa secara langsung dalam penyelesaian tugas akhir ini.
4. Bapak dan Ibu dosen Program Studi Informatika, yang telah memberikan ilmu yang bermanfaat kepada penulis, semoga bapak dan ibu selalu diberikan kesehatan serta lindungan dari Allah SWT.
5. Bapak Rohmi Irsyad, selaku Direktur PT Visi Citra Internasional narasumber utama yang memberikan masukan dalam pengembangan sistem informasi dan selaku pihak travel agen.
6. Dinastutyasakti Bening Kinasih, selaku partner yang memberi semangat dalam pengerjaan tugas akhir.
7. Rekan Madaharsa Yuhdi dan Naufal, selaku tim pengembang perintisan bisnis yang bekerja sama dalam ideasi hingga implementasi.
8. Rekan Harapan Anak Bangsa yang menjadi tempat diskusi dalam penyelesaian tugas akhir ini.

Semoga segala bantuan, dukungan, bimbingan, dan pengajaran yang telah diberikan kepada penulis mendapatkan kebaikan dari Allah SWT. Penulis memohon maaf apabila selama melaksanakan Tugas Akhir terdapat kekhilafan dan kesalahan. Penulis menyadari akan

keterbatasan kemampuan yang dimiliki. Semoga laporan ini dapat bermanfaat bagi semua yang membaca dan menikmatinya.

**Wassalamu'alaikum Warahmatullahi Wabarakatuh**

Yogyakarta, 23 Oktober 2021



(Fardika Rais Hidayatullah)





## SARI

Indonesia merupakan negara dengan muslim terbanyak di dunia. Statistik menunjukkan pada tahun 2010 penduduk Indonesia yang beragama Islam mencapai 209,12 juta jiwa atau sekitar 87% dari total populasi. Pada tahun 2020, penduduk muslim Indonesia diperkirakan akan mencapai 229,62 juta jiwa. Hal ini menjadikan tingkat umroh dan haji di negara Indonesia semakin tinggi. Umroh dan haji merupakan ibadah yang dianjurkan bagi umat muslim yang mampu. Hal tersebut menjadi kesempatan untuk membuat sebuah agen atau biro travel umroh dan haji. Namun, beberapa agen atau biro umroh dan haji masih memiliki cara yang konvensional sehingga membuat beberapa pendataan, maupun pengadaan umroh menjadi rumit dan sulit untuk diatasi. Oleh karena itu Safir hadir sebagai *platform* yang menghubungkan jemaah umroh dan haji dengan agen atau biro umroh dan haji yang telah diverifikasi sehingga dapat menimbulkan rasa kepercayaan kepada jemaah haji untuk melakukan ibadah umroh atau haji. Dalam pengembangan Safir memiliki kendala dalam mengumpulkan kebutuhan pengguna, dikarenakan pandemi yang terjadi menyebabkan banyak travel agen yang tutup dan sulit untuk ditanyai kebutuhannya. Maka untuk mengatasi kebutuhan pengguna yang belum pasti dan dinamis, metode pengembangan yang mendukung adalah *Extreme Programming* (XP). XP dianggap tepat, dan sangat melibatkan pengguna dalam proses pengembangannya. Pemodelan basis data menggunakan *Entity Relationship Diagram* (ERD) sebagai pemberi gambaran basis data. *Tools* pendukung yang digunakan yaitu *Laravel Framework* sebagai *Back-end* dan *Vue.js* sebagai *tools* untuk membangun antarmuka. Safir diharapkan menjadi solusi cepat dan tepat dalam membantu agen atau biro haji dan umroh, serta aman dan nyaman dalam membantu jemaah mewujudkan ibadah haji maupun umroh.

Kata kunci: Safir, *Extreme Programming*, Agen dan Umroh, Jemaah Haji

## GLOSARIUM

<i>Back-End</i>	Segala hal yang berhubungan dengan <i>server</i> dan pertukaran data dengan <i>database</i> .
<i>Framework</i>	Merupakan kerangka kerja untuk mengembangkan perangkat lunak yang bertujuan membantu kinerja pengembangan.
<i>Front-End</i>	Bagian antarmuka yang dapat dilihat langsung oleh pengguna aplikasi dan bisa langsung berinteraksi.
<i>Header Request</i>	Nilai atribut berbasis teks yang ditetapkan oleh <i>client</i> dan <i>server</i> untuk melakukan pertukaran data.
<i>Laravel</i>	Sebuah kerangka kerja aplikasi yang berbasis PHP menggunakan konsep <i>Model-View-Controller</i> .
<i>Open Source</i>	Kode sumber atau kode dasar yang tersedia secara bebas untuk dapat dimodifikasi dan dimanfaatkan.
<i>Request Body</i>	Nilai atribut yang diminta oleh <i>server</i> untuk dapat memberikan data yang sesuai dengan yang diminta.
<i>Requirements</i>	Sebuah kebutuhan yang diminta oleh klien berupa penambahan, perubahan, maupun penghapusan fitur.
<i>Response</i>	Data yang diberikan dari server berdasarkan <i>request</i> yang dikirimkan.
<i>Vue.js</i>	Kerangka kerja antarmuka yang memberikan kemudahan dalam mengatur tata letak, komponen, dan pertukaran data dengan <i>server</i> .
<i>Vuetify</i>	Kerangka kerja yang berlandaskan <i>vue.js</i> dengan fitur untuk mempermudah dalam pembuatan implementasi antarmuka.

## DAFTAR ISI

HALAMAN JUDUL	1
HALAMAN PENGESAHAN DOSEN PEMBIMBING	2
HALAMAN PENGESAHAN DOSEN PENGUJI	3
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	4
HALAMAN PERSEMBAHAN	5
HALAMAN MOTO	6
KATA PENGANTAR	7
SARI	9
GLOSARIUM	10
DAFTAR ISI	11
DAFTAR TABEL	13
DAFTAR GAMBAR	14
BAB I PENDAHULUAN	16
1.1 Latar Belakang	16
1.2 Rumusan Masalah	17
1.3 Batasan Masalah	17
1.4 Tujuan Pengembangan	17
1.5 Manfaat Pengembangan	18
1.6 Metode Pengembangan	18
1.7 Sistematika Penulisan	20
BAB II LANDASAN TEORI	21
2.1 <i>Extreme Programming (XP)</i>	21
2.2 Sistem Informasi	26
2.3 Laravel Framework	26
2.4 Vue.js	28
2.5 Vuetify	30
2.6 REST API	30
BAB III METODOLOGI PENGEMBANGAN	33
3.1 Langkah Pengerjaan Safir	33
3.2 Fase Eksplorasi ( <i>Exploration Phase</i> )	33
3.3 Fase Perencanaan ( <i>Planning Phase</i> )	37
3.4 Iterasi Implementasi ( <i>Iteration to Release Phase</i> )	39
3.4.1 Iterasi Ke-1	39
3.4.2 Iterasi Ke-2	44
3.4.3 Iterasi Ke-3	47
3.4.4 Iterasi Ke-4	49
3.4.5 Iterasi Ke-5	51
3.4.6 Iterasi Ke-6	53
3.5 Aspek Dasar Implementasi Extreme Programming pada Safir	56
3.5.1 <i>Simple Design</i>	57
3.5.2 <i>Refactoring</i>	57
3.5.3 <i>Testing</i>	59
3.5.4 <i>Pair Programming</i>	59
3.5.5 <i>Coding Standards</i>	59
BAB IV HASIL DAN PEMBAHASAN	61
4.1 Hasil	61
4.1.1 Halaman Website Dashboard Admin	61

4.1.2	Halaman Website Dashboard Customer	65
4.1.3	Dokumentasi API <i>Back-End</i>	68
4.2	Pembahasan	73
4.2.1	Kelebihan	73
4.2.2	Kekurangan	74
	BAB V PENUTUP	75
5.1	Kesimpulan	75
5.2	Saran	75
	DAFTAR PUSTAKA	76
2.	LAMPIRAN	77
a.	Basis data Safir	77
b.	Entity Relationship Diagram (ERD) Safir	77
c.	<i>Invoice</i> Pemesanan Safir	78
d.	<i>Dashboard User</i> Safir	79



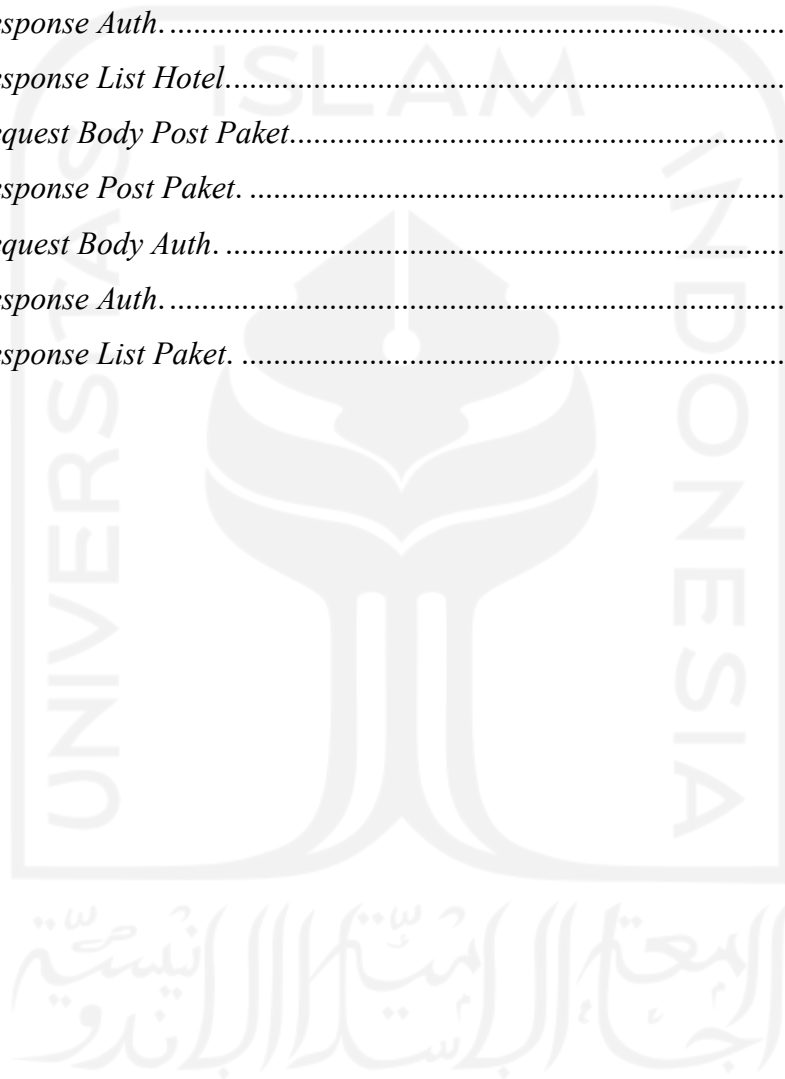
## DAFTAR TABEL

Tabel 2.1 Arsitektur <i>Framework Laravel</i> .....	27
Tabel 3.1 <i>User Stories</i> .....	34
Tabel 3.2 Hasil Analisis Kebutuhan Sistem Karyawan.....	35
Tabel 3.3 Hasil Analisis Kebutuhan Sistem Jemaah .....	36
Tabel 3.4 Deskripsi ERD .....	40
Tabel 3.5 Skema Pengujian Modul Operasional.....	44
Tabel 3.6 Dokumentasi API Paket.....	45
Tabel 3.7 Skema Pengujian Modul Transaksi .....	47
Tabel 3.8 Dokumentasi API Transaksi. ....	47
Tabel 3.9 Skema Pengujian Modul Gudang .....	49
Tabel 3.10 Dokumentasi API Inventaris.....	50
Tabel 3.11 Skema Pengujian Modul Admin.....	51
Tabel 3.12 Dokumentasi API Admins. ....	52
Tabel 3.13 Skema Pengujian Modul User .....	53
Tabel 3.14 Dokumentasi API User. ....	54
Tabel 4.1 API Auth.....	68
Tabel 4.2 API Get List Hotel .....	68
Tabel 4.3 API POST Hotel .....	70
Tabel 4.4 API Auth User .....	71
Tabel 4.5 API Get List Paket .....	72

## DAFTAR GAMBAR

Gambar 2.1 <i>Life Cycle</i> proses XP .....	22
Gambar 2.2 Aspek Dasar XP .....	23
Gambar 2.3 Laravel .....	27
Gambar 2.4 Vue.js .....	28
Gambar 2.5 Vuetify .....	30
Gambar 2.6 REST API .....	31
Gambar 3.1 Langkah Pengerjaan Safir .....	33
Gambar 3.2 <i>Use Case Diagram</i> .....	37
Gambar 3.4 <i>Entity Relationship Diagram</i> (ERD).....	40
Gambar 3.5 Skema Relasi.....	43
Gambar 3.6 Komponen Modul Paket .....	45
Gambar 3.7 Potongan Kode Model Paket.....	46
Gambar 3.8 Komponen Modul Transaksi.....	48
Gambar 3.9 Potongan Kode Model Transaksi.....	48
Gambar 3.10 Komponen Modul Inventaris .....	50
Gambar 3.11 Potongan Kode Model Inventaris .....	50
Gambar 3.12 Komponen Modul Admin .....	52
Gambar 3.13 Potongan Kode Model Inventaris .....	52
Gambar 3.14 Komponen Modul User.....	55
Gambar 3.15 Potongan Kode Modul User.....	55
Gambar 3.16 Tampilan Antar Muka <i>List Paket</i> .....	57
Gambar 3.17 Potongan Fungsi Sebelum <i>Refactoring</i> .....	58
Gambar 3.18 Potongan Fungsi Setelah <i>Refactoring</i> .....	59
Gambar 3.19 Contoh Potongan Kode <i>OperationalService.php</i> .....	60
Gambar 3.20 Contoh Potongan Kode <i>OperationalRepository.php</i> .....	60
Gambar 4.1 Tampilan Halaman Login .....	61
Gambar 4.2 Tampilan Halaman Dashboard.....	62
Gambar 4.3 Tampilan Halaman Paket .....	62
Gambar 4.4 Tampilan Halaman Detail Paket .....	63
Gambar 4.5 Tampilan Halaman Detail Tipe Paket.....	63
Gambar 4.6 Tampilan Halaman Detail Galeri Paket .....	64
Gambar 4.7 Tampilan Halaman Maskapai .....	64

Gambar 4.8 Tampilan Halaman Hotel .....	65
Gambar 4.9 Tampilan Halaman Depan Kustomer.....	66
Gambar 4.10 Tampilan Halaman List Paket.....	66
Gambar 4.11 Tampilan Halaman Detail Paket .....	67
Gambar 4.12 Tampilan Halaman Pemesanan Paket .....	67
Gambar 4.13 Tampilan Halaman Detail Pemesanan .....	67
Gambar 4.14 <i>Request Body Auth.</i> .....	68
Gambar 4.15 <i>Response Auth.</i> .....	68
Gambar 4.16 <i>Response List Hotel</i> .....	69
Gambar 4.17 <i>Request Body Post Paket</i> .....	70
Gambar 4.18 <i>Response Post Paket.</i> .....	71
Gambar 4.19 <i>Request Body Auth.</i> .....	71
Gambar 4.20 <i>Response Auth.</i> .....	71
Gambar 4.21 <i>Response List Paket.</i> .....	73



## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Indonesia merupakan negara muslim terbesar di dunia. Berdasarkan data dari Direktur Jendral Kependudukan dan Pencatatan Sipil Kementerian Dalam Negeri (Dirjen Dukcapil Kemendragi), pada 2021 penduduk Indonesia mencapai 272 juta jiwa. Berdasarkan data dari katadata.co.id, jumlah umat muslim di Indonesia memiliki jumlah yang stabil dari tiap tahunnya pada tahun 2018 jumlah jemaah mencapai 200.000 orang dan pada tahun 2019 sebanyak 212.730 orang. Menurut Asosiasi Muslim Penyelenggara Haji dan Umrah Republik Indonesia (AMPHURI), mencatat pada tahun 1437 H atau 2015-2016 jumlah jemaah hampir menyentuh 700.000 orang. Pada tahun 1439H atau 2017-2018 jumlah jemaah umrah sebanyak 1.000.000. Jumlah ini meningkat sekitar 30% dibandingkan 3 tahun sebelumnya.

Aktivitas Penyelenggara Haji dan Umrah yang beragam berawal dari pemasaran, pendataan jamaah, pencatatan berkas, pendataan kebutuhan jamaah haji dan umrah, mengatur jadwal, hingga keberangkatan jemaah melaksanakan ibadah masih memiliki kekurangan dalam menjalankan aktivitasnya. Masih banyak aktivitas yang dilakukan dengan cara konvensional dan manual. Hal ini menyebabkan timbulnya kesalahan dalam setiap aktivitas yang dilakukan oleh Penyelenggara Haji dan Umrah.

Oleh karena itu, tim pengembangan perintisan bisnis Madaharsa membuat sebuah platform sistem informasi manajemen berbasis website yang diberi nama Safir dengan harapan dapat menyelesaikan permasalahan tersebut. Dalam perintisannya, Safir terdampak oleh pandemi Covid-19 yang menyebabkan banyak bisnis travel tutup. Hal ini menjadi kendala terhadap pengembangan dan kebutuhan sistem karena minimnya informasi terkait alur bisnis dan alur sistem yang terjadi.

Ketidakpastian dalam pengembangan bisnis menjadi salah satu masalah yang harus diselesaikan dalam pengembangan Safir. Proses bisnis yang dinamis dengan kebutuhan yang tidak pasti menjadikan tantangan dalam pengembangan Safir. Agar dapat menangani kebutuhan yang tidak pasti, maka dibutuhkan sebuah metode pengembangan yang dapat beradaptasi dengan perubahan tersebut. Metode yang digunakan adalah *Extreme Programming (XP)* yang dirasa tepat dalam memudahkan pengembangan Safir.

Metode pengembangan *Extreme Programming (XP)* merupakan model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan



tersebut menjadi lebih adaptif dan fleksibel (Widhiarta, 2012). Hal ini menjadi alasan utama dalam penggunaan XP sebagai metode pengembangan yang digunakan untuk dapat mengatasi setiap perubahan kebutuhan yang terjadi. Untuk meningkatkan proses efisiensi pengembangan, maka pengembang menggunakan *tools Laravel Framework* sebagai pembuat API, dan *Vue.js* sebagai antarmuka. Kedua *tools* tersebut menjadi pilihan karena komunitas yang besar sehingga memudahkan dalam diskusi. Serta *Use Case Diagram* dan *Entity Relationship Diagram* sebagai dokumen acuan untuk pengembangan sistem.

## 1.2 Rumusan Masalah

Bagaimana mengembangkan sistem informasi manajemen haji dan umrah berbasis web menggunakan framework Laravel yang dapat memudahkan aktivitas penyelenggara haji dan umrah, serta membantu jemaah haji dan umrah dalam melaksanakan ibadahnya melalui pemanfaatan teknologi dengan metode pengembangan *Extreme Programming (XP)*. Dari latar belakang tersebut maka perumusan masalah sebagai berikut:

- a. Bagaimana merancang Sistem Informasi Manajemen Haji dan Umrah berbasis *website*?
- b. Bagaimana membangun Sistem Informasi Manajemen Haji dan Umrah berbasis *website* dengan menggunakan *restful API*?
- c. Bagaimana mengimplementasikan metode pengembangan *Extreme Programming (XP)* dalam mengembangkan Sistem Informasi Manajemen Haji dan Umrah berbasis *website*?

## 1.3 Batasan Masalah

Agar penelitian lebih terfokus dan tidak meluas dari pembahasan, maka skripsi ini membataskan ruang lingkup penelitian sebagai berikut:

- a. Menggunakan metode pengembangan *Extreme Programming (XP)*.
- b. Luas lingkup sistem yang dibangun meliputi fitur manajemen paket tour, manajemen jemaah, manajemen transaksi, dan manajemen admin.
- c. Penerapan implementasi pengembangan menjadi Fase Eksplorasi, Fase Perencanaan, Fase Implementasi, Fase Produksi, Fase Akhir, dan Fase Perawatan.

## 1.4 Tujuan Pengembangan

Tujuan dari pengembangan ini adalah mengembangkan sistem informasi manajemen haji dan umrah berbasis web dengan menggunakan framework Laravel yang diharapkan memberikan kemudahan bagi penyelenggara haji dan umrah dalam melakukan aktivitasnya dan mampu

membantu jemaah haji dan umrah dalam melaksanakan ibadahnya melalui pemanfaatan teknologi dengan metode pengembangan *Extreme Programming* (XP) yang dianggap tepat dan sesuai berdasar pada memenuhi keinginan klien yang dapat berubah setiap waktu, sedangkan pemodelan basis data digambarkan dengan *Entity Relationship Diagram* (ERD).

### 1.5 Manfaat Pengembangan

Pengembangan ini diharapkan mampu memberikan manfaat bagi penyelenggara haji dan umrah maupun jemaah haji dan umrah di Indonesia sebagai berikut:

- a. Mengembangkan sistem informasi Safir melalui metode pengembangan *Extreme Programming* (XP) sebagai solusi bagi penyelenggara ibadah haji dan umrah dan jemaah di Indonesia untuk saling terhubung dan percaya.
- b. Membantu dalam pelayanan penyelenggara haji dan umrah dalam proses ibadah haji dan umrah meliputi, perencanaan, penjadwalan, pendataan jemaah, hingga pada keberangkatan jemaah.
- c. Turut serta dalam meningkatkan indeks kepuasan jemaah haji dan umrah agar setiap umat muslim di Indonesia dapat menjalankan ibadah haji dan umrah dengan aman, nyaman, dan mudah-mudahan menjadi ibadah yang sempurna.
- d. Meningkatkan animo dan minat masyarakat di Indonesia terhadap haji dan umrah melalui peningkatan kinerja serta pelayanan dari penyelenggara haji dan umrah.

### 1.6 Metode Pengembangan

Metode yang digunakan dalam pengembangan sistem informasi Safir adalah *Extreme Programming* (XP) yang mana terdiri dari beberapa fase dan tahapan, diantaranya:

- a. Fase Eksplorasi

Pada tahapan ini pengembang bertemu dengan pengguna untuk mencari informasi terkait alur bisnis dan kebutuhan pengguna. Pengguna memberikan catatan kebutuhan dari sistem yang mendasar. Kebutuhan tersebut dibuat sederhana untuk dapat memberikan pemahaman kedua pihak terkait sistem yang hendak dibangun.

Fase eksplorasi bertujuan untuk mengetahui keinginan pengguna sehingga dapat menghasilkan ruang lingkup pekerjaan yang lebih terarah. Hasil dari fase ini merupakan *user story* atau gambaran kasar yang diceritakan oleh pengguna yang kemudian akan dijadikan dokumen untuk perencanaan.

- b. Fase Perencanaan

Setelah mendapatkan dokumentasi terkait keinginan pengguna yang telah didapat. Fase ini berorientasi pada fase eksplorasi sebelumnya. Fase ini bertujuan untuk memecahkan dan memetakan kebutuhan pengguna, kebutuhan aspek bisnis, dan kebutuhan sistem yang hendak dibangun. Fase ini memberikan gambaran estimasi perencanaan dalam pembangunan sistem. Fase ini membuat rencana untuk implementasi sistem pada fase selanjutnya. Setiap kebutuhan pengguna diberikan poin kepentingan atau prioritas untuk menentukan tingkat kesulitan ataupun kompleksitas sistem yang hendak dikembangkan. Pada fase ini menghasilkan *use case* diagram untuk memberikan gambaran kebutuhan kepada pengguna dan menentukan kebutuhan sistem seperti perangkat lunak yang dibutuhkan, maupun perangkat yang mendukung sistem tersebut.

c. Iterasi Pengembangan Sistem (*Develop*)

Fase ini merupakan fase implementasi dari dokumen yang telah dibuat sebelumnya menjadi suatu sistem informasi. Pada fase ini terjadi beberapa kali pengulangan atau iterasi yang setiap perulangan atau iterasi tersebut terdiri dari 4 tahapan yaitu analisa, desain, perencanaan pengujian, dan pengujian atau testing. Pada fase ini, dilakukan pengembangan sistem berbasis web menggunakan *Laravel framework* dengan menggunakan *tools* VSCode sebagai *code editor*.

d. Fase Produksi

Pada fase produksi sistem dirilis dengan skala kecil yang bertujuan untuk melihat pengalaman pengguna dalam menggunakan sistem tersebut dan mendapatkan *feedback* atau umpan balik dari pengalaman pengguna. Hasil dari fase ini berupa masukan dari pengguna yang akan dikembalikan ke fase perancangan untuk diimplementasikan lebih lanjut.

e. *Death Phase*

*Death phase* merupakan fase untuk sistem siap dirilis dengan skala besar atau digunakan di pengguna secara umum yaitu pada semua travel yang ada. Fase ini merupakan akhir dari implementasi sistem setelah berbagai pengujian dan masukan pengguna sehingga menjadikan sistem tersebut layak untuk dipasarkan dan digunakan dengan skala yang besar.

f. Fase Pemeliharaan

Fase pemeliharaan merupakan fase terakhir dari semua fase yang telah dilewati. Fase ini lebih memfokuskan pada kelangsungan sistem untuk jangka panjang. Ada beberapa hal yang tetap harus dijaga agar sistem dapat terus berjalan hingga kedepan. Seperti

memastikan cadangan data, memberikan pengamanan sistem, maupun menaikkan versi perangkat lunak pendukung sistem. Meskipun fase pemeliharaan merupakan fase terakhir, tetapi fase ini tidak bisa dianggap remeh dikarenakan fase ini memperhatikan alur data sistem maupun kegiatan transaksi pada suatu sistem.

## 1.7 Sistematika Penulisan

Berisi susunan bab dan subbab pada keseluruhan laporan. Sistematika Penulisan disusun untuk memberikan gambaran umum tentang laporan akhir yang dikerjakan. Sistematika Penulisan adalah sebagai berikut:

a. **BAB I: PENDAHULUAN**

Bab ini berisikan latar belakang pengembangan sistem informasi, rumusan masalah, batasan masalah, tujuan pengembangan, manfaat pengembangan, metode pengembangan, serta sistematika penulisan terkait sistem informasi manajemen haji dan umroh berbasis website menggunakan *framework* Laravel dengan metode *Extreme Programming* (XP) untuk Safir.

b. **BAB II: LANDASAN TEORI**

Bab ini berisi teori-teori yang berhubungan dengan pengembangan dan menjadi landasan dalam pembuatan sistem serta yang berkaitan dengan pengembangan Safir.

c. **BAB III: METODOLOGI PENGEMBANGAN**

Bab ini berisi penjelasan metode yang digunakan dalam pengembangan Safir yaitu *Extreme Programming* (XP) beserta tahapan dan deskripsi dari setiap fase yang dilalui.

d. **BAB IV: HASIL DAN PEMBAHASAN**

Bab ini berisi uraian terkait hasil dari pengembangan yang dicapai beserta kelebihan dan kekurangan pengembangan dengan menggunakan metode yang digunakan serta dokumentasi dari hasil tersebut.

e. **BAB V: PENUTUP**

Bab ini berisi kesimpulan dan saran terhadap pengembangan yang sudah dilakukan dengan tujuan memberikan ringkasan terhadap pengembangan sistem informasi Safir. Kemudian, digunakan untuk dokumentasi di masa yang akan datang.

## BAB II LANDASAN TEORI

### 2.1 *Extreme Programming (XP)*

*Extreme Programming (XP)* pertama kali diterapkan pada proyek pengembangan C3 (*Chrysler Comprehensive Compensation*) dari Chrysler. Proyek ini digunakan untuk menggaji karyawan Chrysler yang memiliki *class* sekitar 2.000 dan 30.000 *method*. Proyek ini terancam gagal karena rumitnya proyek yang dibangun dan pengujian yang gagal. Hal ini yang membuat Chrysler menyewa Kent Beck yang merupakan ahli *software engineering* yang saat ini dikenal sebagai pencetus awal XP. Beck bersama rekannya Ron Jeffries melakukan perubahan pada proyek tersebut untuk membuatnya semakin efisien, fleksibel, dan adaptif. Hal yang menjadi fokus mereka adalah untuk merilis proyek tersebut dalam waktu tidak lebih dari 2 tahun sejak saat dikontrak. Hingga pada akhirnya Beck dan Jeffries berhasil menyelesaikan target dengan menerapkan berbagai metode dalam pengembangan proyek tersebut. Metode-metode inilah yang kemudian mulai dikenal sebagai model pengembangan *Extreme Programming (XP)*.

XP merupakan suatu model pengembangan perangkat lunak yang menyederhanakan tahapan-tahapan dalam pengembangan menjadi lebih adaptif dan fleksibel (Widhiarta, 2012). Metodologi pengembangan XP ditujukan untuk tim yang berukuran kecil hingga medium berkisar 10 hingga 20 orang. Tim dengan jumlah anggota yang kecil bertujuan untuk memenuhi kebutuhan klien yang belum matang maupun adanya perubahan kebutuhan klien yang sangat cepat. XP tidak selalu cocok untuk diterapkan pada setiap proyek. XP memiliki kelebihan untuk dapat menyesuaikan proyek yang memiliki *dynamic requirements* atau proyek dengan kebutuhan klien yang dinamis. Sehingga pengembangan menggunakan XP dituntut untuk dapat beradaptasi dengan cepat dalam mengatasi perubahan yang terjadi selama proses pengembangan proyek.

Sebagai suatu metode pengembangan perangkat lunak yang dinamis, XP memiliki lima *values* yang menjadi nilai-nilai dasar yang diperlukan dalam menggunakan metode pengembangan XP (Widodo & Subekti, 2006). Lima *values* ini adalah:

a. Komunikasi (*Communication*)

Hubungan komunikasi yang baik antar tim menjadi fokus utama oleh XP. Tim harus membangun rasa pengertian dan toleransi serta wajib untuk berbagi pengetahuan dan keterampilan dalam pengembangan.

b. Kesederhanaan (*Simplicity*)

XP menekankan kesederhanaan, menghilangkan fitur-fitur yang tidak terlalu dibutuhkan, menyederhanakan desain, dan berbagai proses lainnya. Menggunakan *method* atau fungsi yang pendek dan sederhana, tidak rumit. Proses penyederhanaan ini akan selalu menjadi nilai utama dari setiap aspek XP.

c. Umpan Balik (*Feedback*)

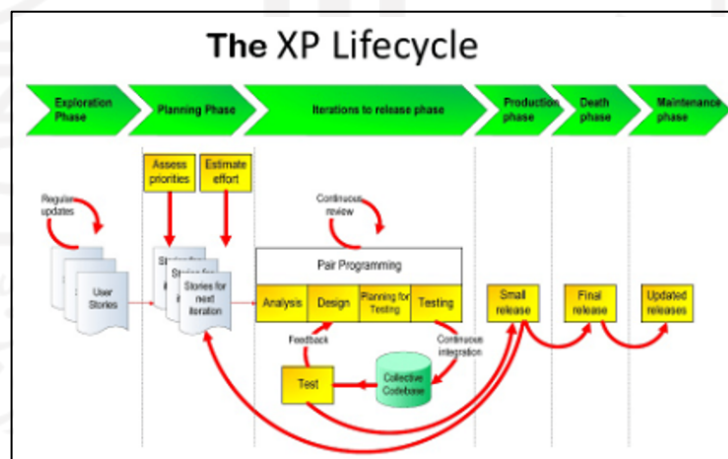
Selalu berikan respon atau *feedback* dari pihak yang terlibat dalam pengembangan. Utarakan setiap pikiran dan berdiskusi setiap masalah yang muncul pada saat pengembangan. Tujuan *feedback* tersebut adalah agar dapat menyadari kesalahan yang dapat ditingkatkan kembali.

d. Keteguhan Hati (*Courage*)

Setiap pihak yang terlibat dalam pengembangan harus memiliki keteguhan hati yang kuat meskipun sedang dalam tekanan dari situasi di sekitar. Agar dapat meneguhkan hati dan berintegritas, setiap anggota tim harus saling percaya satu sama lain. Rasa percaya sesama anggota ini yang ditanam dan ditumbuhkan oleh XP pada setiap aspeknya.

e. Pekerjaan yang Berkualitas (*Quality Work*)

Keempat poin sebelumnya menghasilkan kondisi dimana kita harus melakukan pekerjaan yang maksimal agar dapat memberikan kualitas yang terbaik.

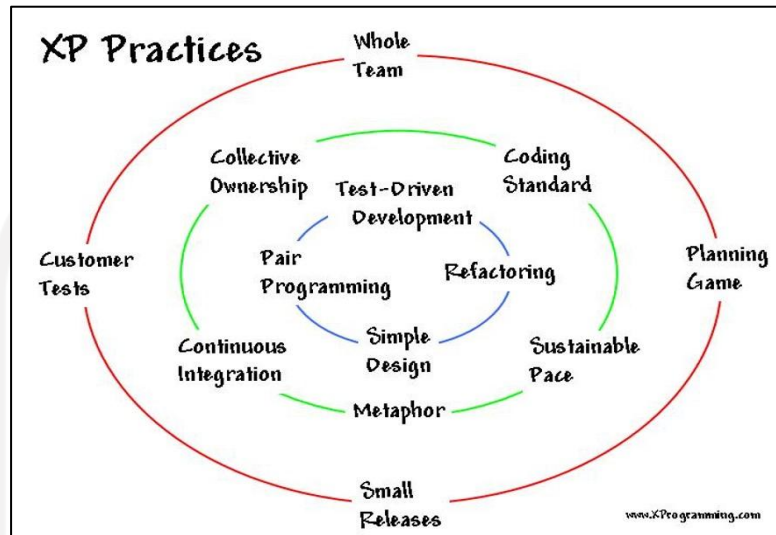


Gambar 2.1 *Life Cycle* proses XP

Sumber: Abrahamsson (2002)

XP sebagai sebuah metodologi tentu memiliki *life cycle* atau siklus hidup. XP memiliki siklus hidup yang terdiri dari 6 fase seperti pada Gambar 2.1 yaitu: Fase Eksplorasi (*Exploration Phase*), Perencanaan (*Planning Phase*), Iterasi Implementasi (*Iterations to Release Phase*), Fase

Produksi (*Production Phase*), Fase Kematian (*Death Phase*), dan Fase Pemeliharaan (*Maintenance Phase*). Dalam membangun XP, Beck dan Jeffries menerapkan berbagai teknik atau metode yang digunakan pada *C3 Project*. Terdapat 12 aspek dasar pada XP yang dapat diamati pada Gambar 2.2.



Gambar 2.2 Aspek Dasar XP

Sumber: XProgramming.com (2021)

Semua aspek dasar menjadi prinsip yang dikembangkan dalam XP. Aspek dasar tersebut memiliki penjelasan antara lain:

a. *The Planning Game*

Pada perencanaan pada XP menggunakan terminologi “*game*”, karena pada penerapannya menerapkan sistem skor dalam menentukan *requirements*. *Requirements* yang semakin sulit maka semakin tinggi skor pada perencanaan tersebut.

b. *Small Release*

Setiap pengembang selesai menyelesaikan suatu modul atau fitur dari perangkat lunak yang dikembangkan maka hasil tersebut harus dipresentasikan secepatnya untuk didiskusikan dengan klien. Bahkan dapat langsung diimplementasikan pada perusahaan klien apabila memungkinkan untuk menerapkannya. Hal ini bertujuan untuk menjadi tes awal dari keseluruhan sistem. Namun tidak selalu untuk melakukan implementasi tersebut setelah mempertimbangkan sumber daya yang dibutuhkan. Apakah lebih menguntungkan langsung melakukan tes atau melakukan. Tes setelah modul tersebut terintegrasi dengan sempurna.

c. *Metaphor*

*Metaphor* pada dasarnya memiliki kesamaan dengan arsitektur perangkat lunak dengan tujuan menggambarkan visi terhadap tujuan dari pengembangan perangkat lunak. Beck menginginkan menyederhanakan proses pengembangan yang saat ini dianggap terlalu rumit. Arsitektur yang berisi diagram dan kode menjadi kendala untuk dimengerti oleh klien. *Metaphor* lebih dapat dimengerti oleh klien dikarenakan bersifat naratif dan deskriptif. Dengan demikian, adanya *Metaphor* dapat membangun komunikasi dengan klien menjadi lebih baik dan lancar.

d. *Simple Design*

Pada XP lebih mengunggulkan keberlangsungan fitur dan modul yang dikembangkan. Sehingga mengusung desain yang tidak terlalu rumit dan sederhana. Penerapan desain simple tersebut bertujuan untuk mengatasi perubahan dan meminimalisir kegagalan desain.

e. *Refactoring*

*Refactoring* merupakan aspek yang khas bagi XP. Menurut Martin Fowler, *Refactoring* adalah perubahan pada kode program dari perangkat lunak dengan tujuan meningkatkan kualitas dari struktur program tersebut tanpa mengubah cara program tersebut bekerja. *Refactoring* menjadi bagian yang paling sesuai dalam XP karena merubah struktur kode menjadi lebih sederhana sehingga meningkatkan kualitas program.

f. *Testing*

XP memiliki perbedaan dengan metode pengembangan lainnya. Pada metode lain, pengujian dilakukan setelah pengembangan selesai dilakukan. Metodologi XP pengembang membuat skenario *testing* yang ingin dijalankan. *Testing* akan jauh lebih baik apabila dilakukan pada setiap modul maupun fungsi yang dikembangkan dalam lingkup sekecil mungkin daripada menunggu sampai keseluruhan sistem selesai dikembangkan.

g. *Pair Programming*

*Pair Programming* adalah melakukan proses pengembangan dengan menulis program secara berpasangan. Penulisan program tersebut dilakukan pada komputer yang sama untuk menyelesaikan suatu modul maupun fungsi. Tujuan dilakukannya ini untuk meningkatkan komunikasi tim pengembang dan berdiskusi serta mengoreksi pekerjaan yang dilakukan. Dalam tahapan awal pengembangan, pengembang tidak menggunakan *pair programming*. Maka perubahan yang dilakukan adalah, mengubah diskusi yang terjadi antar programmer yang terlibat, menjadi diskusi pada forum dan tim pada safir.



Sehingga dalam tahapan pengembangan dan implementasi kode tetap sesuai dengan yang diinginkan.

h. *Collective Ownership*

Kepemilikan program merupakan milik bersama sehingga tidak ada satu baris kode program yang hanya dimengerti satu orang. XP mengharuskan semua *programmer* untuk berbagi pengetahuan mereka terkait program yang dituliskan. Dengan pemahaman yang sama dapat meminimalisir kendala perbedaan penulisan program.

i. *Coding Standards*

*Pair Programming* dan *Collective Ownership* dapat berjalan dengan baik apabila *programmer* dapat memahami kode program dengan baik. Maka dari itu, *Coding Standards* diperlukan untuk menentukan struktur program yang dikembangkan seperti penulisan fungsi maupun variabel.

j. *Continuous Integration*

XP menginginkan untuk dapat melakukan *Build* sesering mungkin. Hal ini bertujuan untuk mengetahui kesalahan maupun *error* yang dapat diperbaiki dengan secepat mungkin. Kebanyakan model pengembangan melakukan *build* sehari sekali. Namun pada XP *build* dilakukan setiap 4 jam atau lebih cepat.

k. *40-hours Week*

Menurut Beck bekerja 8 jam sehari dan 5 hari seminggu menjadi Batasan bagi manusia melakukan pekerjaannya. Jika melebihi itu dapat menyebabkan tingkat *human error* yang lebih tinggi. Sehingga pengembang lebih sering melakukan kesalahan karena terlalu merasa lelah.

l. *On-Site Customer*

XP menganjurkan agar klien sangat terlibat dalam proses pengembangan. Bahkan klien harus ada di tempat pengembangan untuk ikut serta dalam proses *build* maupun pengujian. Adanya klien diharapkan dapat langsung memberikan koreksi pada saat terjadi kesalahan dalam pengembangan.

Aspek dasar memiliki nilai yang tinggi dalam pengembangan perangkat lunak menggunakan metode XP. Semua aspek yang dikembangkan oleh Beck menjadi prinsip yang harus ditanam dan dijalankan agar XP dapat memudahkan dan membuat pengembangan menjadi lebih efektif dan efisien.

## 2.2 Sistem Informasi

Pada era modern saat ini serta digitalisasi pada saat ini, banyak kumpulan sistem yang terdapat di sekeliling kita. Sistem pendaftaran, sistem pemesanan makanan, sistem perbankan dan sebagainya menjadi contoh sistem yang ada di dunia. Sistem merupakan kumpulan elemen yang berhubungan dengan suatu target atau tujuan yang sama untuk dapat menuju tujuan yang ingin dicapai (Firman et al., 2016).

Informasi adalah sekumpulan data yang dimasukkan dan telah diproses menjadi keluaran yang lebih berguna dan dapat dimengerti penggunaannya. Informasi juga dapat berarti suatu data yang telah diklasifikasikan atau dikelompokkan untuk dapat berguna dalam pengambilan keputusan. Sehingga informasi berfungsi untuk memberikan wawasan pada penggunaannya. Suatu sistem tanpa informasi tidak akan memiliki manfaat yang berarti, karena sistem hanya fokus untuk mencapai tujuan tanpa mengetahui informasi untuk mencapai tujuan tersebut (Sutabri, 2012).

Sistem informasi adalah sistem yang menghubungkan pengolah transaksi yang memberi dukungan fungsi suatu organisasi maupun perusahaan yang mengandung kebutuhan manajemen organisasi maupun perusahaan dengan tujuan strategi organisasi atau perusahaan agar dapat memberikan suatu informasi tertentu kepada pihak diluar organisasi maupun perusahaan tersebut. Adapun pengertian lain sistem informasi yaitu sekumpulan komponen pada organisasi maupun perusahaan yang berkaitan dengan proses pembuatan dan pemberian informasi (Indrajit, 2000).

## 2.3 Laravel Framework

Laravel adalah framework PHP yang kuat, dirancang untuk pengembang yang membutuhkan toolkit sederhana dan elegan untuk membuat aplikasi web berfitur lengkap. Laravel dibuat oleh Taylor Otwell dan bersifat *open-source*. Laravel ditujukan untuk pengembangan aplikasi berbasis web mengikuti pola arsitektur *model-view-controller* (MVC). Pola arsitektur MVC bertujuan untuk memenuhi *single responsibility principle* (SRP). Prinsip ini merupakan pemisahan komponen berdasarkan tugasnya masing-masing sehingga Laravel dapat membagi ketiga tugas tersebut pada aplikasi (Yadav et al., 2019). Implementasi laravel pada sistem Safir hanya memanfaatkan *model* dan *controller* untuk membuat API sebagai *back-end* aplikasi.



Gambar 2.3 Laravel  
Sumber: laravel.com (2021)

Beberapa fitur dari Laravel adalah sistem pengemasan modular (*modular packaging system*) dengan manajer ketergantungan khusus atau disebut sebagai *dedicated dependency manager*. Kerangka kerja laravel mudah dimengerti dan kuat, kerangka itu sendiri terdiri atas *authentication, routing, session manager, caching* dan banyak komponen yang paling umum digunakan (Laravel.com, 2021). Kemudian, terdapat *database migration tools* dan dukungan pengujian unit terintegrasi, semua alat ini membantu dan memberikan seorang pengembang/*developer* kemampuan untuk membangun sebuah aplikasi yang kompleks. Pengerjaan aplikasi dengan *framework* Laravel dimulai dengan menyiapkan *tools* yang dibutuhkan seperti MySQL sebagai *database, Composer, Database Client Tools* seperti PHPMyAdmin atau DataGrip, dan memilih Sublime Text, VSCode atau PHPStorm sebagai *Code Editor*.

Laravel terdiri dari beberapa komponen penting seperti *Routing, Eloquent ORM Support, Migrations, Blade Templating, Artisan* dan *Bundles* (McCool, 2012). Secara rinci, penjelasan arsitektur dari framework Laravel terdapat pada Tabel 2.1.

Tabel 2.1 Arsitektur *Framework Laravel*

No	Arsitektur Framework Laravel	Keterangan
1	<i>Routing</i>	Digunakan untuk mengelola URL pada aplikasi
2	<i>Eloquent ORM (object-relational-mapping) Support</i>	<i>Developer</i> dipermudah dengan ORM untuk melakukan transaksi ke <i>database</i> seperti CRUD ( <i>Create, Read, Update, Delete</i> ).
3	<i>Migrations</i>	<i>Developer</i> dapat melakukan perubahan dan desain untuk <i>database</i> pada sebuah <i>platform</i> secara independen sebab implementasi <i>database scheme</i> direpresentasikan dalam sebuah <i>class</i> .
4	<i>Blade Templating</i>	Berfungsi untuk menggabungkan data variable dan template ke dalam sebuah tampilan yang disimpan pada cache Laravel. Selain itu, template ini mampu menjalankan loop, conditional yang termasuk dalam fungsi PHP.

5	<i>Bundles</i>	Menyediakan <i>package system</i> yang sifatnya modular. Di Laravel 4, terdapat penambahan Composer pada Laravel yang semakin mempermudah untuk menambahkan modul-modul sesuai dengan <i>package</i> yang tersedia pada <i>repository packagist</i> .
6	<i>Artisan</i>	Mempermudah <i>developer</i> berinteraksi dengan aplikasi yang dikembangkan menggunakan terminal untuk menjalankan <i>unit test</i> , <i>migration</i> , <i>scheduled task</i> , serta membuat sebuah <i>controller</i> .

## 2.4 Vue.js

Vue.js merupakan *framework* JavaScript yang sifatnya progresif *open-source* yang digunakan untuk mengembangkan antarmuka/*interface* web secara interaktif. Vue.js adalah salah satu kerangka kerja yang terkenal dengan penyederhanaan pengembangan web. Vue.js berfokus pada lapisan tampilan dan dapat dengan mudah diintegrasikan ke dalam proyek-proyek besar untuk pengembangan *front-end* (Preston, 2018).



Gambar 2.4 Vue.js

Sumber: v3.vuejs.org (2021)

Vue.js memiliki fitur-fitur yang mendukung penggunaan *framework* ini antara lain sebagai berikut:

a. *Virtual DOM*

Vue.js memanfaatkan DOM virtual, yang juga digunakan oleh kerangka kerja lain seperti React, Ember, dll. Perubahan tidak dilakukan pada DOM, melainkan dibuat replika DOM yang hadir dalam bentuk struktur data JavaScript. Setiap kali ada perubahan yang akan dibuat, mereka dibuat pada struktur data JavaScript dan yang terakhir dibandingkan dengan struktur data asli. Perubahan terakhir kemudian

diperbarui ke DOM asli, yang akan dilihat pengguna berubah. Ini bagus dalam hal pengoptimalan, lebih murah dan perubahan dapat dilakukan lebih cepat.

b. *Data Binding*

Fitur pengikatan data membantu memanipulasi atau menetapkan nilai ke atribut HTML, mengubah gaya, menetapkan *class* dengan bantuan pengikatan yang disebut v-bind tersedia dengan Vue.js.

c. *Components*

*Components* adalah salah satu fitur penting Vue.js yang membantu membuat elemen kustom, yang dapat digunakan kembali dalam HTML. *Components* komponen merupakan elemen khusus yang dilampirkan oleh *compiler* Vue.

d. *Transition*

Vue menyediakan berbagai cara untuk menerapkan efek transisi ketika item dimasukkan, diperbarui, atau dihapus dari DOM. *Transition* pada Vue termasuk tools yang digunakan untuk:

- Secara otomatis menerapkan *class* untuk CSS *transition* dan *animation*.
- Terintegrasi *library* CSS *animation* pihak ketiga seperti Animate.css
- Menggunakan JavaScript untuk memanipulasi DOM secara langsung selama transisi.
- Terintegrasi *library* JavaScript *animation*, seperti Velocity.js.

e. *Computed Properties*

Salah satu fitur penting dari Vue.js untuk melihat perubahan yang dibuat pada elemen UI.

f. *Templates*

Vue.js menyediakan *template* berbasis HTML yang mengikat DOM dengan data instance Vue.

g. *Routing*

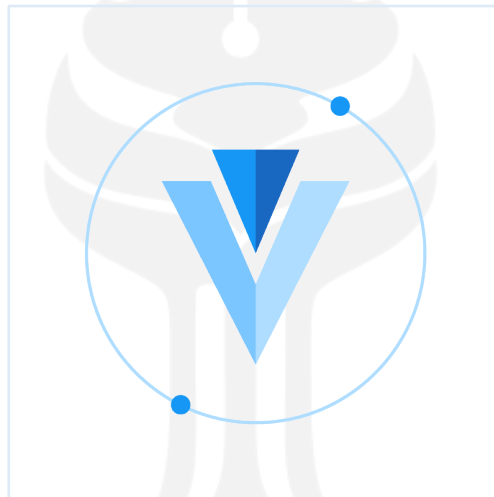
Vue menyediakan antarmuka untuk mengubah apa yang ditampilkan pada halaman berdasarkan jalur URL. Selain itu, navigasi antar halaman dilakukan dengan bantuan vue-router.

Pemanfaatan Vue.js pada sistem Safir menggunakan Vuetify sebagai *framework* yang mempermudah dalam implementasi desain antarmuka.

## 2.5 Vuetify

Vuetify adalah *framework* UI lengkap yang dibangun di atas Vue.js. Tujuan dari proyek ini adalah untuk menyediakan sebuah *tools* untuk pengembang dalam membangun *user experience* yang kaya dan menarik. Tidak seperti kerangka kerja lainnya, Vuetify dirancang agar mudah dipelajari dan mudah untuk memahami ratusan komponen yang dikembangkan oleh *Material Design Specification* (Sharo, 2020).

Vuetify mengambil pendekatan pertama untuk mendesain yang berarti aplikasi Anda hanya berfungsi pada ponsel, tablet, atau komputer desktop. Vuetify dikembangkan persis sesuai dengan *Material Design Specification* dengan setiap komponen yang dibuat cermat agar menjadi modular, responsif, dan berkinerja. Tata Letak yang unik, dinamis dan gaya komponen menggunakan variabel SASS dapat disesuaikan dengan aplikasi yang akan dan atau sedang dikembangkan (Vuetifyjs.com, 2021).



Gambar 2.5 Vuetify

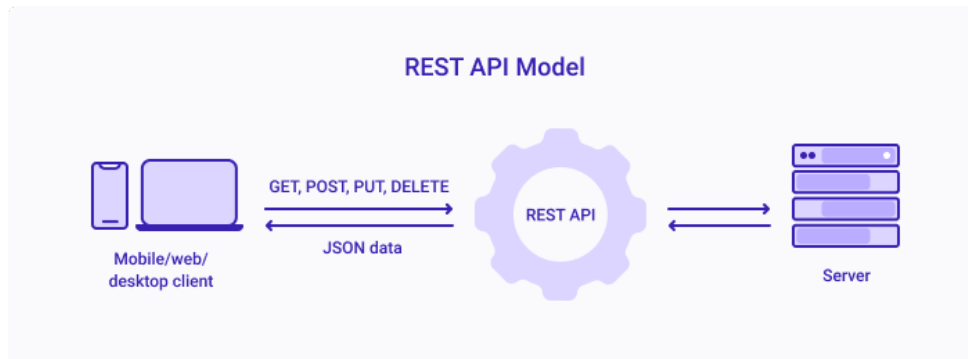
Sumber: vuetifyjs.com (2021)

Vuetify memiliki siklus pengembangan yang sangat aktif, menanggapi masalah dan laporan komunitas dengan kecepatan sangat tinggi, memungkinkan untuk lebih sering mendapatkan perbaikan *bug* dan peningkatan. Selain itu, setiap rilis mayor disertai dengan dukungan jangka panjang selama 18 bulan untuk versi minor sebelumnya.

## 2.6 REST API

*Representational State Transfer* (REST) atau layanan web RESTful adalah cara menyediakan interoperabilitas antara sistem komputer di internet. Layanan Web yang sesuai dengan REST memungkinkan sistem yang meminta untuk mengakses dan memanipulasi

representasi tekstual dari sumber daya web menggunakan serangkaian operasi *stateless* yang beragam dan telah ditentukan sebelumnya (Chen et al., 2017). REST diciptakan oleh Roy Fielding untuk menggambarkan pola desain untuk mengimplementasikan sistem jaringan.



Gambar 2.6 REST API

Sumber: voximplant.com (2021)

Dalam arsitektur REST, klien mengirimkan permintaan pesan menggunakan metode HTTP standar. Struktur kerangka kerja ini memungkinkan skalabilitas yang lebih besar, format data standar bersama dengan integritas data, semua menggunakan protokol permintaan pesan yang aman. Klien dapat berupa aplikasi web, aplikasi seluler, atau pada dasarnya sistem komputer apa pun yang memerlukan akses ke data yang terdapat dalam sistem pihak ketiga.

Format pertukaran data umum yang digunakan dalam *REST API* termasuk HTTP, HTTPS, JSON, HTML, XML, dan banyak lagi. Layanan web RESTful dapat digunakan untuk menghubungkan satu klien atau aplikasi ke arsitektur sistem perangkat lunak yang tidak perlu dipertimbangkan klien saat mengakses data. Semua pemformatan dan standarisasi diprogram ke dalam arsitektur REST API (iron.io, 2020). Tujuan utama *REST* adalah untuk meningkatkan properti non-fungsional yang berbeda dari sistem web, seperti kinerja, skalabilitas, kesederhanaan, keandalan, visibilitas dan lain-lain (Banias et al., 2021).

*REST API* menggunakan apa yang dikenal sebagai operasi *Create, Read, Update, Delete* (CRUD). Keempat fungsi ini adalah permintaan yang dibuat ke server oleh klien, menggunakan metode HTTP standar *GET, POST, PUT, UPDATE, dan DELETE*. Layanan web RESTful dibuat pada antarmuka komputer yang memungkinkan klien untuk melakukan permintaan data yang telah ditentukan sebelumnya. Karakteristik penting dari *REST* adalah bahwa sisi server tetap tanpa status di antara beberapa interaksi, setiap server dalam cluster dapat melayani klien pada setiap permintaan (Chen et al., 2017).

## 2.7 Blackbox Testing

Dalam tahapan pengembangan, pengujian dilakukan untuk memastikan hasil yang didapat dari sistem sesuai dengan yang diinginkan. Dalam pengembangan menggunakan metode *Extreme Programming* (XP), pengujian menjadi fokus yang harus dilakukan dalam setiap iterasi guna meminimalisir terjadinya kesalahan dan penyimpangan dalam implementasi sistem yang dikembangkan. Pada pengujian metode yang digunakan yaitu *Blackbox Testing*.

*Blackbox Testing* merupakan teknik pengujian perangkat lunak yang berfokus pada fungsional dari perangkat lunak (S. Roohullah Jan, S. Tauhid Ullah Shah., 2016). Metode pengujian ini hanya focus pada keluaran yang dihasilkan terhadap masukan dan kondisi eksekusi yang dipilih (Nidhra, S., & Dondeti, J., 2012).

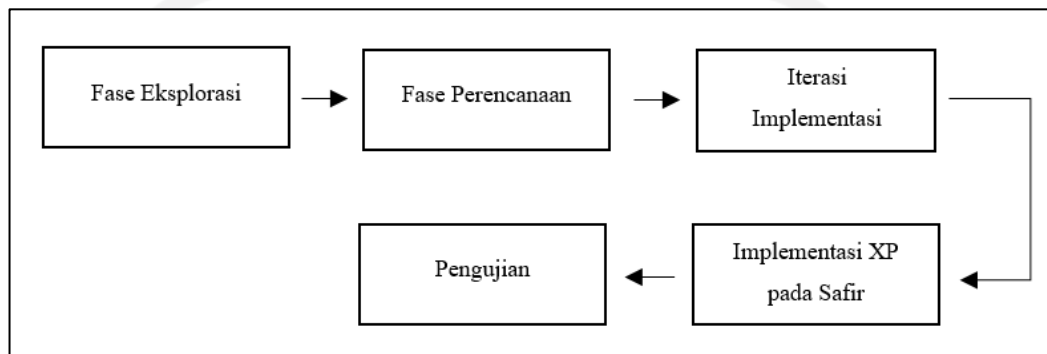




## BAB III METODOLOGI PENGEMBANGAN

### 3.1 Langkah Pengerjaan Safir

Gambar 3.1 menunjukkan langkah-langkah yang digunakan dalam mengerjakan sistem Safir, yaitu fase eksplorasi, fase perencanaan, iterasi implementasi, implelementasi metode XP dan pengujian.



Gambar 3.1 Langkah Pengerjaan Safir

### 3.2 Fase Eksplorasi (*Exploration Phase*)

Pada fase eksplorasi diperlukan pertemuan dengan pengguna untuk mengidentifikasi kebutuhan dari sistem yang ingin dikembangkan. Dilakukan analisis oleh tim Safir yang terdiri dari 3 orang yakni Fardika Rais Hidayatullah selaku ketua tim yang bertugas mengambil keputusan, mengorganisir tim dalam perencanaan pengembangan agar dapat sesuai dengan kebutuhan yang diinginkan dan target waktu yang ditentukan, sekaligus sebagai *Hacker* yang bertanggung jawab pada pengembangan sistem dari analisis perancangan sistem hingga implementasi sistem *website* untuk Safir. Sementara anggota tim lainnya yakni Rohmat Yuhdi bertindak sebagai *Hustler* yang bertanggung jawab dalam ideasi dan melakukan riset pasar dan Naufal Alfiansyah Kurniawan sebagai *Hipster* yang bertanggung jawab pada desain antarmuka Safir.

Pengguna Safir merupakan karyawan agen travel haji dan umrah. Pertemuan dengan pihak agen travel haji bertujuan untuk mendapatkan gambaran proses bisnis, alur transaksi, permasalahan yang terjadi dan alur operasional yang dilakukan oleh agen travel. Setelah itu dibuatlah sebuah *user stories* untuk mengidentifikasi aktor atau *role* pengguna, dan interaksi terhadap sistem. Tabel 3.1 merupakan *user stories* yang digunakan dalam pengembangan Safir.

Tabel 3.1 *User Stories*

No	User	Role	Kebutuhan Sistem
1	Karyawan	Keuangan	Keuangan dapat melihat daftar transaksi dari Jemaah yang melakukan pembayaran dengan rincian sebagai berikut: <ol style="list-style-type: none"> <li>Total harga transaksi</li> <li>Jumlah yang terbayarkan</li> <li>Jumlah yang belum terbayarkan</li> </ol>
			Keuangan dapat melakukan verifikasi pembayaran. Setelah pembayaran dikonfirmasi, konsumen mendapatkan notifikasi bahwa pembayaran diterima.
			Rincian terkait transaksi dapat di <i>export</i> oleh pihak Keuangan ke bentuk excel dengan format <i>file .xlsx</i> yang nantinya akan digunakan sebagai laporan/pembukuan di sistem Safir.
		Operasional	Operasional dapat membuat paket tour baru.
			Operasional dapat mengatur paket tour yang sudah ada
			Operasional dapat melihat pendaftar pada paket yang telah diinputkan.
			Operasional dapat melihat kelengkapan dokumen dari konsumen.
			Operasional dapat membuat jadwal agenda yang diperlukan untuk pra keberangkatan. Jadwal tersebut berupa: <ol style="list-style-type: none"> <li>Pengambilan kelengkapan konsumen</li> <li>Pengarahan kepada konsumen</li> </ol>
			Operasional dapat melihat transaksi untuk memberi tau kepada konsumen terkait pembayaran yang telah dilakukan maupun belum dilakukan.
		Gudang	Gudang dapat memasukkan data opname dari kesediaan inventaris.
			Gudang dapat menginputkan konsumen yang telah menerima kelengkapan konsumen.
		Super Admin	Super admin dapat melakukan semua aktifitas pada staff lainnya.
			Super admin dapat membuat akun baru untuk staff.
			Super admin dapat mengatur akun staff.
		2	Konsumen
Konsumen dapat melihat detail paket tour yang tersedia			
Konsumen dapat mendaftar ke paket yang dipilih dan mendapatkan notifikasi pendaftaran.			
Konsumen dapat menginput bukti pembayaran ke sistem.			
Konsumen dapat melihat rekap pembayaran			
Konsumen dapat melihat agenda yang telah dijadwalkan oleh agen travel.			

			Konsumen harus melakukan “Login” untuk dapat melakukan pendaftaran dan transaksi.
			Konsumen dapat melakukan pendaftaran diri.

Berdasarkan *user stories* yang tercantum dalam Tabel 3.1, pengguna sistem dibagi menjadi 2 kategori yaitu Karyawan dan Konsumen. Karyawan memiliki divisi keuangan, operasional dan gudang serta super admin sebagai aktor yang mengatur keseluruhan akun staff. Melalui *user stories* yang didapat dari pengguna, Safir memiliki fitur-fitur yang dibuat. Manajemen Tour, Manajemen Jemaah, Manajemen Transaksi, Manajemen Inventaris, dan Pendaftaran Jemaah menjadi fitur utama. Dari *user stories* yang dibuat, tim dapat membuat desain rancangan antarmuka sistem, dan struktur data yang digunakan dalam pengembangan sistem Safir.

Setelah pembuatan *user stories*, dibuat perancangan antarmuka untuk sistem Safir dengan melakukan analisis kebutuhan sistem bagi karyawan dan konsumen (Jemaah). Hasil analisis kebutuhan sistem karyawan terdapat pada Tabel 3.2.

Tabel 3.2 Hasil Analisis Kebutuhan Sistem Karyawan

No	Halaman Karyawan	Fungsi yang Disediakan
1	Halaman Muka	Menampilkan grafik transaksi harian, maupun bulanan.
		Menampilkan jumlah paket yang berjalan, jumlah jemaah yang terdaftar, dan konfirmasi pembayaran.
2	Halaman Paket	Menampilkan seluruh daftar paket yang telah didaftarkan sebelumnya.
		Terdapat menu <i>filter</i> yang berguna untuk mensortir paket terdekat, berdasarkan abjad, jumlah kuota, dan paket terbaru.
		Menampilkan fungsi “Edit”, dan “Detail”.
3	Halaman Pembuatan Paket	Menampilkan <i>form</i> isian data paket yang diinputkan.
		Menampilkan <i>list</i> tipe paket.
		Menampilkan <i>list</i> hotel.
		Menampilkan <i>list</i> maskapai.
4	Halaman Data Jemaah	Menampilkan <i>list</i> paket yang aktif.
		Menampilkan fungsi aksi “detail”, “pembayaran”, “inventaris”.
		Ketika aksi “detail” ditekan, menampilkan <i>list</i> jemaah yang telah terdaftar dalam paket tersebut.
		Pada tampilan detail jemaah, menampilkan data jemaah, riwayat pembayaran, dan detail berkas yang diserahkan oleh jemaah.
		Pada tampilan aksi “inventaris” menampilkan detail jumlah barang yang akan digunakan dalam paket tersebut.
		Terdapat aksi fungsi “Tambah” untuk menampilkan form isian data inventaris dan “edit” untuk merubah data inventaris yang telah dimasukkan.
5	Halaman Transaksi	Menampilkan info transaksi keseluruhan.

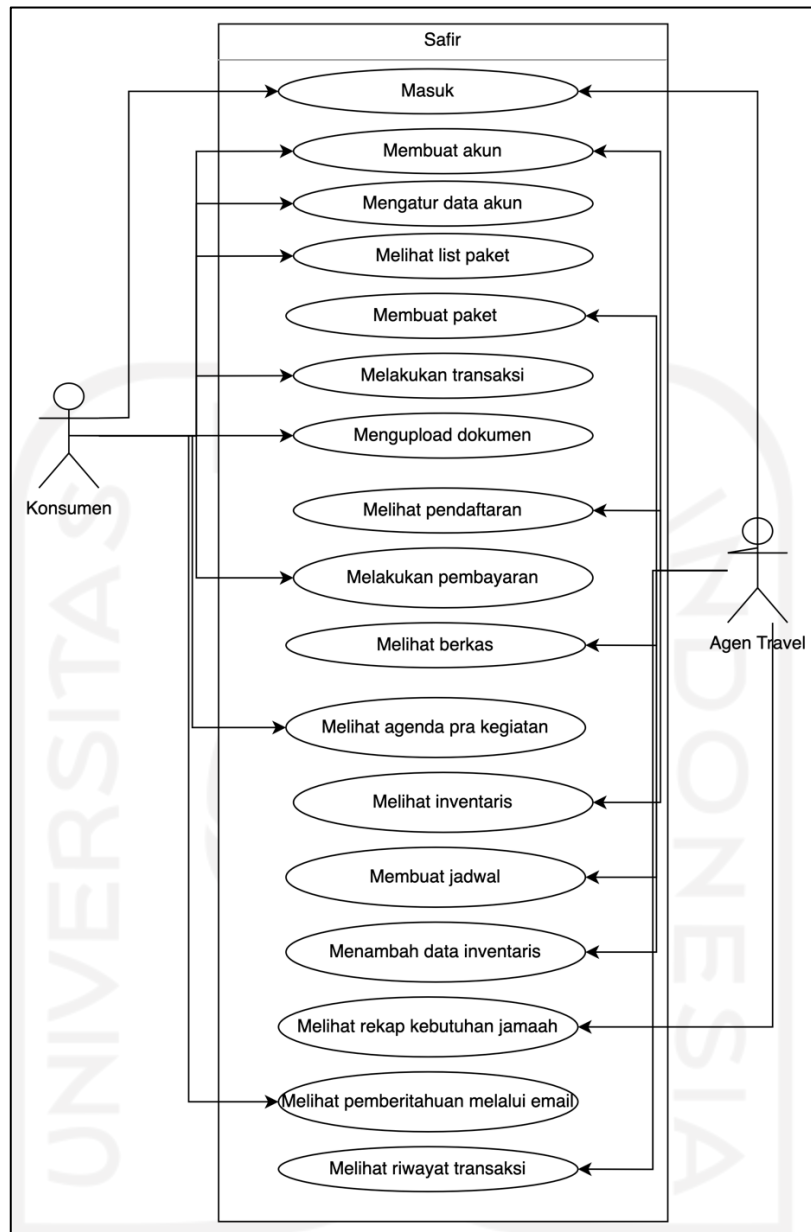
		Menampilkan fungsi aksi “detail” pada setiap transaksi untuk menampilkan detail info dari transaksi tersebut.
		Terdapat fitur <i>filter</i> untuk mengurutkan transaksi terbaru, transaksi berdasarkan nominal, dan pencarian data transaksi.
		Menginginkan untuk terdapat menu ekspor ke dokumen untuk pembukuan.

Hasil analisis kebutuhan sistem jemaah terdapat pada Tabel 3.3.

Tabel 3.3 Hasil Analisis Kebutuhan Sistem Jemaah

No	Halaman Jemaah	Fungsi yang Disediakan
1	Halaman Muka	Menampilkan daftar paket yang dapat didaftarkan oleh jemaah.
2	Halaman Paket	Menampilkan detail paket yang dipilih. Kemudian terdapat fungsi aksi “Pesan Paket” untuk mendaftar paket tersebut dan diarahkan ke menu pendaftaran
3	Halaman Pendaftaran	Menampilkan form isian berupa data diri jemaah dan pemesanan untuk jemaah
4	Halaman Pembayaran	Menampilkan form isian untuk memasukan pembayaran. Untuk pembayaran, jemaah memasukkan kode transaksi dan kode pemesanan yang telah diberikan sebelumnya.
5	Halaman Riwayat Pembayaran	Menampilkan list riwayat penmbayaran yang telah dibayarkan sebelumnya.
6	Halaman Detail Paket	Menginginkan untuk terdapat tombol <i>download</i> atau unduh agar dapat mendownload paket dalam bentuk poster.

Selain *user stories* dan analisis kebutuhan sistem telah dirancang, terdapat penggambaran *use case diagram* untuk pengguna sistem yaitu Karyawan dan Konsumen (Jemaah). *Use case diagram* Karyawan menggambarkan kebutuhan karyawan yang terdiri dari masuk, mendaftar, membuat paket, melihat pendaftaran, melihat berkas, melihat riwayat transaksi, melihat inventaris, menambah data inventaris, membuat jadwal, dan melihat rekap kebutuhan jemaah. *Use case diagram* Karyawan dapat dilihat pada Gambar 3.2.



Gambar 3.2 Use Case Diagram

*Use case diagram* Konsumen (Jemaah) menggambarkan kebutuhan jemaah yaitu terdiri dari mendaftar, melihat paket, melihat detail paket, mendaftar paket, mengirim berkas, melihat kelengkapan berkas, melihat riwayat pembayaran, dan melihat agenda pra kegiatan. *Use case diagram* Jemaah terdapat pada Gambar 3.2.

### 3.3 Fase Perencanaan (*Planning Phase*)

Setelah *user stories* dibuat, maka pada fase ini bertujuan untuk merencanakan iterasi yang akan diimplementasikan dalam pengembangan. Berdasarkan *user stories* pengembangan dibagi menjadi beberapa iterasi. Pembagian ini bertujuan untuk menyelesaikan modul-modul dari fitur

menjadi lebih kecil agar dapat menjadi fokus dalam pengembangan dan meminimalisir kesalahan dari sistem maupun koordinasi dengan agen travel. Pada fase ini, iterasi yang akan dilakukan dibagi menjadi 6 iterasi dengan rincian sebagai berikut:

a. Pembuatan Struktur Data

Sebelum melakukan implementasi modul, dibutuhkan suatu rancangan basis data yang akan digunakan dalam pengembangan. Struktur data ini akan saling memiliki keterkaitan pada setiap entitasnya. Pembuatan struktur data menggunakan *tools* dbdiagram.io yang memudahkan dalam pembuatan struktur data tersebut. Setelah struktur data selesai dibuat, maka struktur data tersebut dapat di ekspor menjadi basis data *SQL* yang dapat digunakan dalam pengembangan.

b. Modul Operasional

Modul Operasional menjadi tahapan pertama dalam implementasi sistem. Modul operasional berisi pengaturan data Hotel dan Maskapai, penambahan paket baru, perubahan paket, maupun rekap paket yang dipesan oleh customer atau Jemaah.

Modul ini menjadi modul utama yang harus diselesaikan karena data yang dihasilkan akan sangat memiliki keterkaitan dengan modul yang lain seperti modul transaksi, dan modul inventaris. Pengerjaan modul dimulai dari pengerjaan *back-end* seperti pembuatan model, kontroler, hingga menjadikan API yang dapat digunakan pada pengerjaan front-end meliputi implementasi antarmuka dan implementasi API dalam pertukaran data.

c. Modul Transaksi

Setelah Modul Operational selesai dikerjakan, implementasi selanjutnya adalah Modul Transaksi. Modul ini berisi rekap transaksi, pembuatan tagihan, verifikasi transfer dan pembayaran, serta rekap pembayaran yang telah dilakukan oleh konsumen atau Jemaah. Pengerjaan modul transaksksi dimulai dari pembuatan model yang menghubungkan dengan model pada modul operasional kemudian menghasilkan API yang digunakan untuk *front-end*.

d. Modul Inventaris

Modul Inventaris dibuat untuk menunjang pendataan opname barang yang dimiliki oleh pihak travel agen. Modul ini berisi pendataan barang, maupun rekap penambahan dan pengurangan barang. Pengerjaan modul ini akan menghubungkan

model pada modul operasional dan modul transaksi, sehingga pendataan barang bagi consumer atau Jemaah akan tercatat oleh sistem.

e. Modul Admin

Modul admin dibuat untuk mengatur akun yang akan digunakan oleh karyawan dari travel agen. Modul admin ini berisi pembuatan akun, perubahan akun, maupun penghapusan akun. Modul ini pada dasarnya tidak berkaitan dengan modul yang lain tetapi menjadi catatan penting dalam setiap tindakan yang dilakukan oleh karyawan dalam setiap aktivitas yang dilakukan pada sistem.

f. Modul Transaksi bagi Konsumen

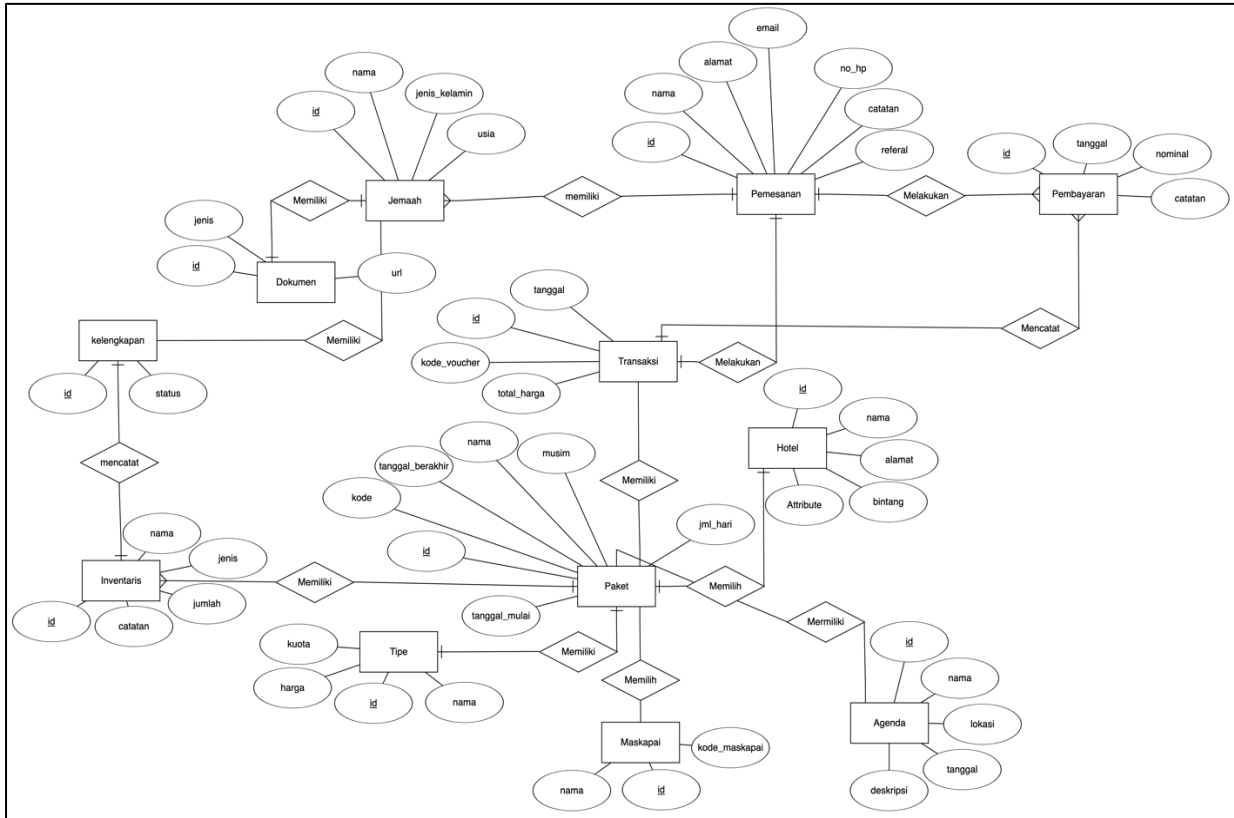
Pengerjaan modul ini menggunakan data dari modul transaksi yang telah dibuat. Modul ini akan menjadi rekap pembayaran yang dapat dilihat dari sisi konsumen atau Jemaah.

### 3.4 Iterasi Implementasi (*Iteration to Release Phase*)

Pada fase ini dilakukan implementasi sistem berdasarkan *user stories* yang telah dibuat pada fase sebelumnya. Interaksi dengan pengguna diperlukan dalam tahap ini untuk mengoreksi apabila terdapat kesalahan dalam sistem yang dibuat. Pada tahapan pengembangan Safir memiliki 3 orang dalam 1 tim dan hanya memiliki seorang programmer. Dengan demikian, *Pair Programming* pada tahapan iterasi hanya dilakukan oleh seorang programmer dengan bantuan komunitas sebagai pemberi masukan, kritik, dan saran dalam standarisasi pengkodean. Sedangkan untuk mendukung masukan dan kemudahan dalam struktur pengkodean, maka programmer melakukan riset dan diskusi antar programmer untuk mendukung standarisasi pengkodean yang dijalankan.

#### 3.4.1 Iterasi Ke-1

Iterasi pertama pada tahapan ini dibuat gambaran desain basis data yang dibuat berdasarkan identifikasi *user stories* sebelumnya. Pembuatan iterasi ini menggunakan *tools* ERDPlus untuk membuat *Entity Relationship Diagram* (ERD), dan skema relasi tabel basis data.



Gambar 3.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) menghasilkan 12 entitas yang terhubung dengan relasi seperti pada Gambar 3.1. Entitas tersebut memiliki detail yang terdapat pada Tabel 3.4.

Tabel 3.4 Deskripsi ERD

No	Entitas	Atribut
1	JEMAAH	id → bigint nama → varchar jenis_kelamin → varchar usia → int id_pemesanan → bigint id_paket → bigint created_at → timestamp updated_at → timestamp
2	PAKET	id → bigint kode → varchar nama → varchar jml_hari → int tgl_mulai → date tgl_berakhir → date id_hotel → bigint id_maskapai → bigint status → varchar created_at → timestamp



		updated_at → timestamp
3	TIPE	id → bigint nama → varchar harga → double kuota → int id_paket → bigint created_at → timestamp updated_at → timestamp
4	PEMESANAN	id → bigint nama → varchar alamat → text email → varchar no_hp → varchar catatan → text code → varchar status → smallint user_id → bigint id_transaksi → bigint created_at → timestamp updated_at → timestamp
5	TRANSAKSI	id → bigint tanggal → date kode_voucher → varchar total_harga → double sudah_dibayar → double belum_bayar → double created_at → timestamp updated_at → timestamp
6	TAGIHAN	id → bigint id_pemesanan → bigint deskripsi → varchar tipe → varchar limit_at → date nominal → double status → tinyint created_at → timestamp updated_at → timestamp
7	PEMBAYARAN	id → bigint nominal → double catatan → text id_pemesanan → bigint id_tagihan → bigint created_at → timestamp updated_at → timestamp
8	MASKAPAI	id → bigint kode_maskapai → varchar nama → varchar created_at → timestamp updated_at → timestamp
9	HOTEL	id → bigint name → varchar alamat → text

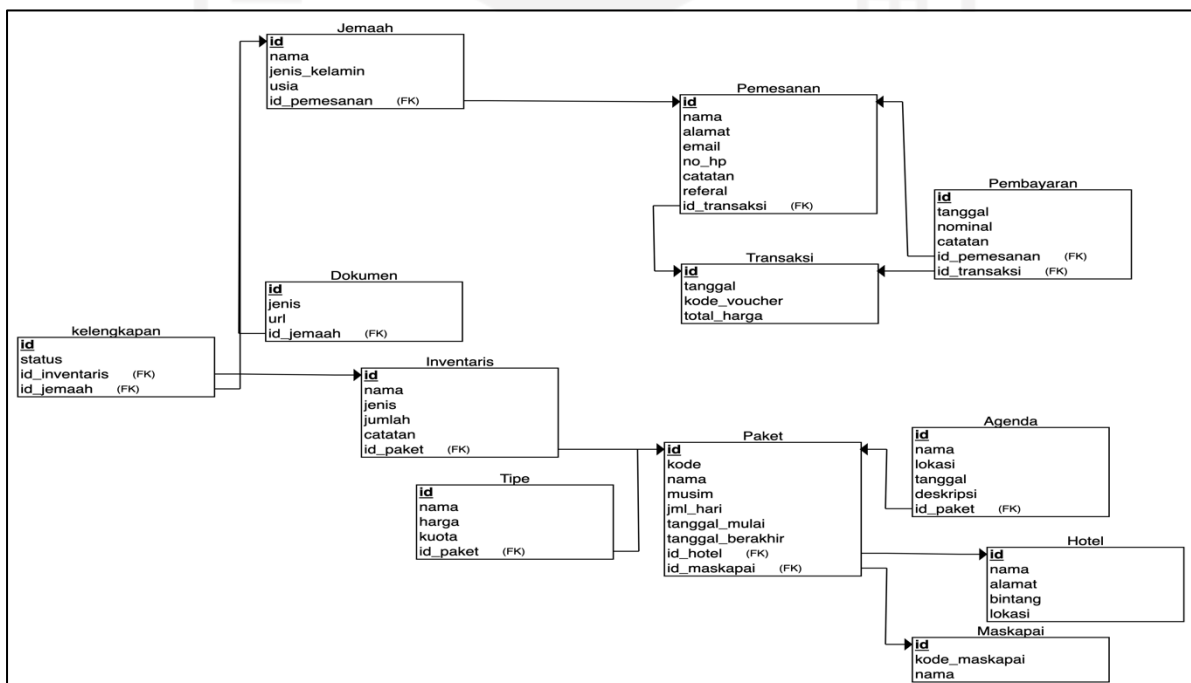
		lokasi → varchar created_at → timestamp updated_at → timestamp
10	INVENTARIS	id → bigint nama → varchar jenis → varchar jumlah → int catatan → text created_at → timestamp updated_at → timestamp
11	KELENGKAPAN	id → bigint status → varchar id_inventaris → bigint id_jemaah → bigint created_at → timestamp updated_at → timestamp
12	AGENDA	id → bigint nama → varchar lokasi → varchar tanggal → date deskripsi → text id_paket → bigint created_at → timestamp updated_at → timestamp

Entitas JEMAAH memiliki atribut id sebagai atribut unik, nama, jenis\_kelamin, usia, entitas ini bertujuan untuk menyimpan data diri jemaah yang didaftarkan. Entitas DOKUMEN memiliki relasi dengan JEMAAH memiliki atribut id sebagai atribut unik, jenis, dan url, entitas ini bertujuan untuk menyimpan dokumen jemaah seperti visa, passport, maupun surat keterangan vaksin. Entitas KELENGKAPAN memiliki relasi dengan Jemaah dan INVENTARIS, id sebagai atribut unik, dan status, entitas ini bertujuan untuk mencatat pendataan kelengkapan jemaah seperti seragam, kain ihram dan koper. Relasi dengan INVENTARIS yang memiliki atribut id sebagai atribut unik, nama, jenis, catatan, dan jumlah, entitas ini bertujuan untuk mendata stok opname barang yang tersedia dan diberikan sebagai kelengkapan jamaah. Setiap Jamaah yang melakukan pendaftaran maka stok barang kelengkapan jemaah yang ada akan langsung berkurang. Entitas PEMESANAN memiliki id sebagai atribut unik, nama, alamat, email, no\_hp, catatan, dan referral, entitas ini bertujuan untuk merekap jemaah yang mendaftar sehingga dalam satu kali pemesanan dapat terdiri lebih dari 1 jemaah. PEMESANAN memiliki relasi dengan JEMAAH untuk mengetahui jamaah tersebut terdaftar atas satu pemesanan dan relasi dengan TRANSAKSI sebagai rekap dari pembayaran yang telah dilakukan dan nominal yang harus dibayarkan, entitas ini memiliki atribut id sebagai atribut unik, tanggal, kode\_voucher, dan total\_harga. TRANSAKSI memiliki relasi dengan PAKET untuk mengidentifikasi paket tour yang dipesan. Entitas ini memiliki id sebagai

atribut unik, kode sebagai kode paket yang dibuat, tanggal\_mulai, tanggal\_akhir sebagai keberangkatan dan kepulangan, dan jml\_hari. Entitas PAKET memiliki relasi dengan TIPE untuk menyimpan jenis paket dari tour yang dibuat, entitas ini memiliki id sebagai atribut unik, kuota, harga, dan nama.

Entitas AGENDA memiliki atribut id sebagai atribut unik, nama, lokasi, tanggal, dan deskripsi, entitas ini digunakan untuk menjadwalkan kegiatan pra-kegiatan yang dilakukan oleh agen travel. AGENDA memiliki relasi dengan PAKET untuk menjadwalkan berdasarkan paket tour dan menginformasikan kepada jemaah yang telah memesan paket tersebut. Entitas HOTEL memiliki id sebagai atribut unik, nama, alamat, bintang, dan lokasi, entitas ini sebagai data hotel yang menjadi rujukan dalam pemilihan hotel untuk paket yang dibuat. Entitas MASKAPAI memiliki atribut id sebagai atribut unik, nama dan kode\_maskapai, entitas ini sebagai daftar maskapai yang menjadi rujukan dalam pemilihan transportasi pesawat dalam pembuatan paket.

Setelah pembuatan ERD, dengan menggunakan *tools* ERDPlus diagram dikonversi menjadi skema relasi yang digunakan untuk memodelkan kedalam basis data beserta penggunaan *key* dalam atribut yang meliputi *primary key* dan *foreign key*. Hasil konversi harus memperhatikan tipe data yang digunakan dalam atribut dari setiap tabel dengan tujuan agar relasi tersebut memiliki atribut yang sesuai untuk diimplementasikan ke dalam sistem.



Gambar 3.4 Skema Relasi

Hasil dari konversi dari ERD menjadi skema relasi seperti terlihat pada Gambar 3.2. Skema relasi ini memudahkan dalam pengembangan dan mempersingkat sehingga tidak memerlukan pembuatan relasi secara manual. Pada relasi terdapat 12 tabel yang sesuai dengan perancangan ERD.

### 3.4.2 Iterasi Ke-2

Pada iterasi kedua memulai untuk implementasi dari hasil perancangan pada iterasi sebelumnya menjadi bentuk sistem. Iterasi kedua memfokuskan pada modul Operasional. Sebelum pengerjaan implementasi sistem, dibutuhkan skenario pengetesan berdasarkan *user stories* yang telah dibuat. Tujuan dari skenario ini agar implementasi sistem sesuai dengan kebutuhan yang diminta oleh *user*. Pengetesan yang dilakukan menggunakan metode *Black-box Testing*.

Tabel 3.5 Skema Pengujian Modul Operasional

Kode	Skenario Pengujian	Kasus Pengujian	Hasil yang Diharapkan
OP1	Melihat daftar paket tour.	Klik menu paket.	Menampilkan daftar paket tour yang telah di inputkan.
OP2	Menambah paket tour	Klik menu paket, kemudian klik tombol tambah paket, masukkan semua data, dan klik tombol tambah.	Data paket tour berhasil ditambah.
OP3	Melihat detail tour	Klik menu paket, kemudian klik tombol icon mata pada daftar paket tour.	Menampilkan detail paket tour yang dipilih.
OP4	Melihat daftar peserta tour	Klik menu paket, kemudian klik tombol icon mata pada daftar, kemudian klik tombol daftar jamaah.	Menampilkan daftar peserta tour yang mengikuti paket tour.
OP5	Melakukan <i>update</i> paket tour	Klik menu paket, kemudian klik icon pensil pada daftar, kemudian isikan data yang ingin dirubah dan klik tombol simpan.	Data paket tour berhasil dirubah.
OP6	Melihat kelengkapan dokumen peserta	Klik menu paket, kemudian klik tombol mata pada daftar, kemudian klik daftar jamaah, dan klik dokumen	Menampilkan daftar <i>checklist</i> kelengkapan dokumen peserta.
OP7	Menambah data agenda	Klik menu paket, kemudian detail paket, kemudian klik Agenda, kemudian klik tombol tambah agenda, masukkan semua data, klik tombol tambah.	Berhasil menambahkan agenda.
OP8	Mengupdate agenda	Klik menu paket, detail paket, dan klik tombol agenda, klik icon pensil pada daftar agenda, masukkan data yang ingin di rubah, klik tombol simpan.	Berhasil mengupdate agenda.

Skenario yang telah dibuat seperti pada Tabel 3.5 menjadi acuan bagi pengembang untuk mengimplementasikan sistem sehingga dapat mempercepat proses implementasi yang dilakukan oleh pengembang. Skenario pengujian akan digunakan untuk pengujian setelah proses iterasi untuk modul Operasional selesai. Tahapan implementasi pada modul ini dilakukan dalam 2 tahapan yaitu:

### 1. Implementasi *Back-End*

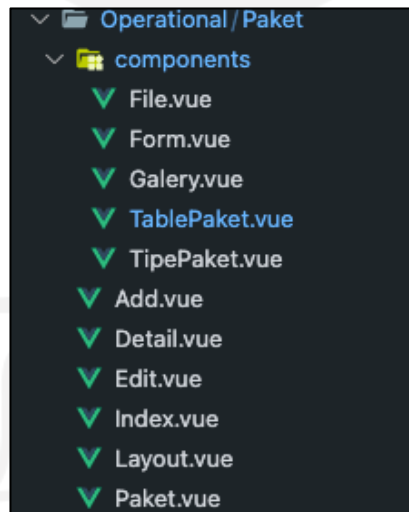
Implementasi *back-end* membuat API untuk modul operasional antara lain seperti terdapat pada Tabel 3.6.

Tabel 3.6 Dokumentasi API Paket.

No	Method	Path	Deskripsi
1	GET	/paket/	Untuk mengambil list paket.
2	GET	/paket/{id_paket}	Untuk mengambil detail paket.
3	POST	/paket/	Untuk menambahkan paket baru.
4	PUT	/paket/{id_paket}	Untuk merubah paket yang tersedia.
5	GET	/paket/{id_paket}/tipe	Untuk melihat tipe paket berdasarkan paket yang dipilih.

### 2. Implementasi *Front-End*

Implementasi *front-end* pada modul operasional membuat komponen yang akan digunakan dalam tampilan antarmuka dan pembuatan model untuk integrasi dengan API.



Gambar 3.5 Komponen Modul Paket

```

*/ Potongan skrip model Paket */
getPackages({ commit, state }, payload) {
  let search = typeof payload !== "undefined" ? payload : "";
  return new Promise((resolve) => {
    $axios
      .get(`/paket?page=${state.page}&q=${search}`)
      .then((response) => {
        commit("ASSIGN_DATA", response.data.data);
      });
  });
}

```

```

        resolve(response.data);
    })
    .catch((error) => {
        if (error.response.status) {
            commit("ASSIGN_DATA", []);
        }
    });
});
},
*/ Potongan skrip model Paket */

```

Gambar 3.6 Potongan Kode Model Paket

### Pair Programming

Pada pelaksanaan *Pair Programming* di iterasi ini, pengembang melakukan implementasi perancangan yang telah dibuat sebelumnya menjadi bentuk kode. Dikarenakan hanya terdapat satu orang *programmer*, maka pelaksanaan *pair programming* yang berpasangan dirubah menjadi *solo programming*. Untuk mendukung *real time problem solving*, setiap implementasi modul dilakukan diskusi langsung dengan tim bisnis untuk mengkonfirmasi pengerjaan agar sesuai. Pengembangan pada iterasi ini, *programmer* melakukan riset terhadap *coding standart* yang akan digunakan. Terutama untuk pengembangan sistem kedepan ketika tim sudah lebih besar.

### Testing

Pengujian dilakukan berdasarkan skema pengujian yang telah dibuat sebelumnya. Hasil dari pengujian sesuai dengan skema yang dibuat dapat dilihat pada tabel 3.7.

Tabel 3.7 Hasil Pengujian Modul Operasional

Kode	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
OP1	Menampilkan daftar paket tour yang telah di inputkan.	Sistem menampilkan list paket tour.	Berhasil
OP2	Data paket tour berhasil ditambah.	Sistem memunculkan notifikasi paket berhasil disimpan.	Berhasil
OP3	Menampilkan detail paket tour yang dipilih.	Sistem menampilkan rincian paket berdasarkan id paket yang dipilih.	Berhasil
OP4	Menampilkan daftar peserta tour yang mengikuti paket tour.	Sistem menampilkan list jemaah yang mengikuti paket yang dipilih.	Berhasil
OP5	Data paket tour berhasil dirubah.	Sistem memunculkan notifikasi paket berhasil diupdate dan data berubah.	Berhasil
OP6	Menampilkan daftar <i>checklist</i> kelengkapan dokumen peserta.	Sistem menampilkan list jemaah berserta <i>checklist</i> dokumen.	Berhasil
OP7	Berhasil menambahkan agenda.	Sistem memunculkan notifikasi agenda berhasil disimpan. Data yang	Berhasil

		dimasukan telah tersedia pada list agenda.	
OP8	Berhasil mengupdate agenda	Sistem memunculkan notifikasi agneda berhasil diupdate.	Berhasil

### 3.4.3 Iterasi Ke-3

Setelah iterasi untuk implementasi modul operasional selesai dilakukan selanjutnya pengembangan difokuskan pada modul transaksi. Modul Transaksi dibutuhkan untuk mencatat setiap transaksi yang terjadi pada sistem sehingga modul ini tetap menjadi fokus utama dalam pengembangan sistem.

Tabel 3.8 Skema Pengujian Modul Transaksi

Kode	Skenario Pengujian	Kasus Pengujian	Hasil yang Diharapkan
TR1	Melihat daftar transaksi	Klik menu transaksi pada menu	Menampilkan daftar transaksi
TR2	Melihat daftar verifikasi pembayaran	Klik menu pembayaran	Menampilkan daftar pembayaran yang belum diverifikasi.
TR3	Memverifikasi pembayaran	Klik menu pembayaran, klik tombol verifikasi, masukkan password, klik tombol submit	Pembayaran berhasil diverifikasi, status berubah, mengupdate transaksi.
TR4	Mengekspor transaksi	Klik menu transaksi, klik menu export	<i>Export</i> menjadi <i>file excel</i> .

Skenario pengujian pada Tabel 3.8 memfokuskan untuk modul transaksi dan pembayaran. Skenario ini akan digunakan untuk menguji modul Transaksi untuk memastikan modul yang dikembangkan sesuai dengan kebutuhan *user*. Tahapan implementasi pada modul ini dilakukan dalam 2 tahapan yaitu:

#### 1. Implementasi *Back-End*

Implementasi *back-end* membuat API untuk modul transaksi antara lain seperti terdapat pada Tabel 3.9.

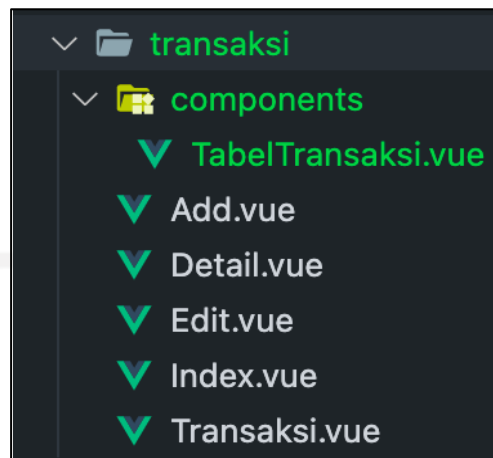
Tabel 3.9 Dokumentasi API Transaksi.

No	<i>Method</i>	<i>Path</i>	Deskripsi
1	GET	/transaksi/	Untuk mengambil list transaksi.
2	GET	/transaksi/{id transaksi}	Untuk mengambil detail transaksi.
3	POST	/transaksi/	Untuk menambahkan transaksi baru.
4	PUT	/transaksi/{id transaksi}	Untuk merubah transaksi yang tersedia.

#### 2. Implementasi *Front-End*

Pada modul transaksi, implementasi *front-end* membuat komponen-komponen yang dibutuhkan pada antarmuka. Salah satunya yaitu komponen TabelTransaksi untuk

menampilkan daftar transaksi yang terdapat pada Gambar 3.8, kemudian untuk integrasi data melalui API terdapat pada Gambar 3.9.



Gambar 3.7 Komponen Modul Transaksi

```

*/ Potongan skrip model Transaksi */
getTransactions({ commit, state }, payload) {
  let search = typeof payload !== "undefined" ? payload : "";
  return new Promise((resolve) => {
    $axios
      .get(`/transaksi?page=${state.page}&q=${search}`)
      .then((response) => {
        commit("ASSIGN_TRANSACTIONS", response.data.data);
        resolve(response.data);
      })
      .catch((error) => {
        if (error.response.status) {
          commit("ASSIGN_TRANSACTIONS", []);
        }
      });
  });
}
*/ Potongan skrip model Transaksi */

```

Gambar 3.8 Potongan Kode Model Transaksi

### *Pair Programming*

Pada iterasi ke-3, penerapan pair programming masih mengikuti pola yang telah dibuat pada iterasi ke-1 dengan *coding standard* yang sama. Dikarenakan hanya memiliki seorang programmer, terdapat kendala dalam implementasi modul transaksi. Terutama pada penerapan alur logika program yang digunakan untuk memverifikasi pembayaran, mendapatkan detail pembayaran, dan mengupdate setiap model ketika ada pembayaran yang dilakukan.

Untuk membantu menyelesaikan kendala tersebut, penulis melakukan diskusi dengan antar programmer dari tim lain untuk membantu penyelesaian modul transaksi. Dari hasil diskusi tersebut, penulis mendapat masukan untuk mengimplementasikan fitur pada *laravel framework* untuk menggunakan *event* dan *listener*. Sehingga ketika ada *event* pembayaran, maka akan



otomatis mengeksekusi *listener* yang disiapkan dan merubah status pada model tabel database yang lainnya.

#### Pengujian

Hasil pengujian dari modul transaksi dapat dilihat pada Tabel 3.10.

Tabel 3.10 Hasil Pengujian Modul Transaksi

Kode	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
TR1	Menampilkan daftar transaksi	Sistem menampilkan list transaksi yang telah berlangsung.	Berhasil
TR2	Menampilkan daftar pembayaran yang belum diverifikasi.	Sistem menampilkan list pembayaran dengan status belum terverifikasi.	Berhasil
TR3	Pembayaran berhasil diverifikasi, status berubah, mengupdate transaksi.	Sistem memunculkan notifikasi pembayaran berhasil diubah dan mengubah status menjadi "Sudah Dibayar"	Berhasil
TR4	<i>Export</i> menjadi <i>file excel</i> .	Sistem melakukan <i>export</i> dan <i>file excel</i> berhasil diunduh.	Berhasil

#### 3.4.4 Iterasi Ke-4

Pada iterasi selanjutnya modul yang diimplementasikan adalah modul Gudang. Modul Gudang bertujuan untuk menyimpan data opname barang yang ada dan mencatat barang yang keluar untuk kelengkapan peserta.

Tabel 3.11 Skema Pengujian Modul Gudang

Kode	Skenario Pengujian	Kasus Pengujian	Hasil yang Diharapkan
GD1	Menambahkan data barang	Klik menu Inventaris, klik tambah barang, masukkan data, klik tombol submit	Data berhasil ditambahkan.
GD2	Mengupdate data barang	Klik menu Inventaris, Klik tombol ikon mata, masukkan data yang ingin diubah, lalu klik submit	Data berhasil diubah.
GD3	Mengupdate kelengkapan peserta	Klik menu Kelengkapan, klik paket yang diikuti peserta, cari nama peserta, masukkan kelengkapan yang diberikan, klik simpan.	Data kelengkapan peserta berhasil diupdate.
GD4	Melihat daftar kelengkapan	Klik menu kelengkapan, klik paket yang dipilih.	Menampilkan daftar peserta dan status kelengkapan peserta.

Skenario pengujian seperti pada Tabel 3.11 digunakan untuk menguji modul Gudang yang memfokuskan terhadap opname barang dan kelengkapan jemaah. Tahapan implementasi pada modul ini dilakukan dalam 2 tahapan yaitu:

## 1. Implementasi *Back-End*

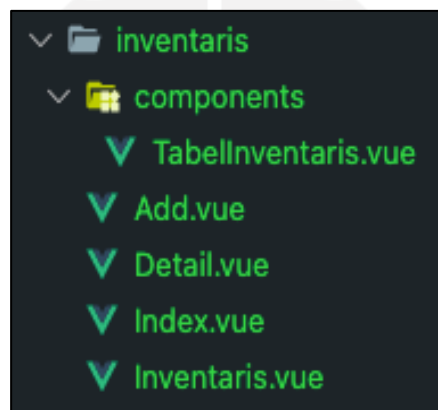
Implementasi back-end adalah membuat API untuk modul Inventaris seperti pada Tabel 3.12.

Tabel 3.12 Dokumentasi API Inventaris.

No	Method	Path	Deskripsi
1	GET	/inventaris/	Untuk mengambil list inventaris.
2	GET	/inventaris/{id inventaris}	Untuk mengabil detail inventaris.
3	POST	/inventaris /	Untuk menambahkan inventaris baru.
4	PUT	/inventaris/{id inventaris}	Untuk merubah inventaris yang tersedia.

## 2. Implementasi *Front-End*

Pada modul Inventaris, tampilan yang dibutuhkan adalah antarmuka untuk menambahkan data barang, merubah, maupun menghapus barang. Pada modul ini dibutuhkan komponen `TabellInventaris` untuk menampilkan list barang yang telah ditambahkan.



Gambar 3.9 Komponen Modul Inventaris

```

*/ Potongan skrip model Inventaris */
getListInventaris({ commit, state }, payload) {
  let search = typeof payload !== "undefined" ? payload : "";
  return new Promise((resolve) => {
    $axios
      .get(`/inventaris?page=${state.page}&q=${search}`)
      .then((response) => {
        commit("ASSIGN_LIST_INVENTARIS", response.data.data);
        resolve(response.data);
      })
      .catch((error) => {
        if (error.response.status) {
          commit("ASSIGN_LIST_INVENTARIS", []);
        }
      });
  });
},
*/ Potongan skrip model Inventaris */

```

Gambar 3.10 Potongan Kode Model Inventaris

### Pair Programming

Dalam iterasi ke-4 pengembangan dilakukan dengan mengikuti *coding standard* yang telah dibuat sebelumnya sehingga memudahkan dalam fase perawatan kedepannya dan memudahkan dalam dokumentasi untuk digunakan dalam tim yang besar. Pada pengembangan Modul Gudang, ada beberapa kendala dalam pengembangannya. Karena kebutuhan dari klien menginginkan untuk dapat mendata setiap kelengkapan Jemaah dan dapat mendata Jemaah yang belum maupun sudah menerima barang. Akan tetapi pendataan Jemaah masih belum dapat menangani kebutuhan tersebut. Maka dalam penerapan fitur tersebut, pengembangan akan dilakukan pada fase selanjutnya.

### Pengujian

Hasil dari pengujian pada fase ke-4 dapat dilihat pada Tabel 3.13.

Tabel 3.13 Hasil Pengujian Modul Gudang

Kode	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
GD1	Data berhasil ditambahkan.	Sistem menampilkan notifikasi “Data berhasil ditambahkan” dan data tersimpan pada <i>database</i> .	Berhasil
GD2	Data berhasil diubah.	Sistem menampilkan notifikasi “Data berhasil diubah” dan data berganti sesuai dengan data yang diganti.	Berhasil
GD3	Data kelengkapan peserta berhasil diupdate.	Sistem belum dapat menangani kelengkapan peserta.	Tidak Berhasil
GD4	Menampilkan daftar peserta dan status kelengkapan peserta.	Sistem belum dapat menangani kelengkapan peserta.	Tidak Berhasil

### 3.4.5 Iterasi Ke-5

Iterasi ke-5 memfokuskan untuk modul Admin. Pada dasarnya modul Admin memiliki fitur untuk dapat melakukan aktivitas semua staff dan aktivitas tambahan berupa manajemen akun staff. Pengerjaan akun admin ini memfokuskan untuk pengaturan akun dikarenakan beberapa modul telah terselesaikan pada iterasi sebelumnya.

Tabel 3.14 Skema Pengujian Modul Admin

Kode	Skenario Pengujian	Kasus Pengujian	Hasil yang Diharapkan
AM1	Menambahkan akun staff	Klik menu Akun, klik tombol tambah akun, masukan data staff, lalu klik tombol tambah.	Data Staff berhasil ditambahkan
AM2	Mengupdate akun staff	Klik menu akun, klik ikon pensil, masukkan data staff yang ingin diubah, klik tombol simpang	Data Staff berhasil diubah.

AM3	Menghapus akun staff	Klik menu akun, klik ikon tempat sampah, klik tombol konfirmasi	Data Staff berhasil dihapus
-----	----------------------	---	-----------------------------

Pada iterasi ini, skenario pengujian lebih difokuskan untuk modul Data Akun Staff. Dikarenakan modul lainnya telah selesai diimplementasikan pada iterasi sebelumnya. Tahapan implementasi pada modul ini dilakukan dalam 2 tahapan yaitu:

### 1. Implementasi *Back-End*

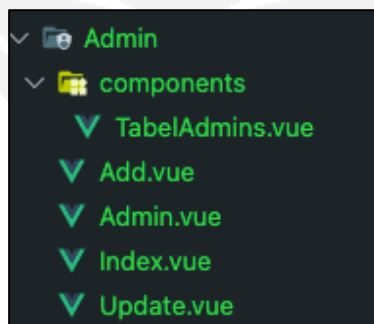
Pada back-end, modul yang dibuat yaitu penambahan admin, perubahan data admin, dan penghapusan admin. Skema API yang dibuat seperti pada Tabel 3.15.

Tabel 3.15 Dokumentasi API Admins.

No	Method	Path	Deskripsi
1	GET	/staff/	Untuk mengambil list admin.
2	GET	/staff/{id_staff}	Untuk mengambil detail admin.
3	POST	/staff/	Untuk menambahkan admin baru.
4	PUT	/staff/{id_staff}	Untuk merubah admin yang tersedia.

### 2. Implementasi *Front-End*

Pada implementasi admin, tampilan yang dibuat membutuhkan tabel untuk list admin. Seperti yang tertera pada Gambar 3.12.



Gambar 3.11 Komponen Modul Admin

```

*/ Potongan skrip model Admin */
getListAdmins({ commit, state }, payload) {
  let search = typeof payload !== "undefined" ? payload : "";
  return new Promise((resolve) => {
    $axios
      .get(`/admins?page=${state.page}&q=${search}`)
      .then((response) => {
        commit("ASSIGN_LIST_ADMINS", response.data.data);
        resolve(response.data);
      })
      .catch((error) => {
        if (error.response.status) {
          commit("ASSIGN_LIST_ADMINS", []);
        }
      });
  });
}
*/ Potongan skrip model Admin */

```

Gambar 3.12 Potongan Kode Model Inventaris

### Pair Programming

Pelaksanaan iterasi ke-5 modul admin menggunakan *coding standard* yang telah dibuat, dengan kerangka yang disusun menggunakan *Repository Design Pattern*. Dalam implementasi modul admin terdapat kendala dalam pemberian *session* dan *token* yang digunakan untuk mengakses dengan tingkatan perizinan yang berbeda-beda. Pada tahap ini, *programmer* menemukan metode autentikasi yang memudahkan dalam pemberian izin *user*. Pemberian *token* dan hak akses menggunakan modul dari *Laravel* yaitu *Passport*. Dengan menggunakan modul ini, *token* akses dapat dibuat sesuai dengan *scope* atau perizinannya masing-masing.

### Pengujian

Hasil dari pengujian sesuai dengan skenario yang telah dibuat dapat dilihat pada Tabel 3.16.

Tabel 3.16 Hasil Pengujian Modul Admin

Kode	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
AM1	Data Staff berhasil ditambahkan	Sistem menampilkan notifikasi "Data berhasil ditambahkan".	Berhasil
AM2	Data Staff berhasil diubah.	Sistem menampilkan notifikasi "Data berhasil disimpan".	Berhasil
AM3	Data Staff berhasil dihapus	Sistem memunculkan konfirmasi "Yakin ingin menghapus data?" dan memunculkan "Data berhasil dihapus" ketika menekan tombol "Ya".	Berhasil

### 3.4.6 Iterasi Ke-6

Iterasi sebelumnya telah menyelesaikan modul-modul yang akan digunakan oleh karyawan agen. Pada iterasi ini pengembangan difokuskan untuk pengembangan modul untuk konsumen. Modul ini digunakan untuk konsumen melakukan pendaftaran dan transaksi pada sistem.

Tabel 3.17 Skema Pengujian Modul User

Kode	Skenario Pengujian	Kasus Pengujian	Hasil yang Diharapkan
MU1	Melakukan <i>Login</i>	Memasukkan data diri, dan klik tombol masuk	Berhasil login dan diarahkan untuk ke beranda.
MU2	Melakukan <i>Register</i>	Klik menu daftar, masukkan data diri, klik tombol daftar	Berhasil mendaftarkan diri dan diarahkan ke beranda dengan status login.
MU3	Melihat Daftar paket	Klik menu beranda	Menampilkan daftar paket yang tersedia.
MU4	Melihat detail paket	Klik menu beranda, klik salah satu paket	Menampilkan detail paket yang tersedia.
MU5	Mendaftar paket	Klik menu beranda, klik salah satu paket, pilih tipe paket dan masukan jumlah peserta,	Berhasil melakukan pendaftaran, sistem akan memberikan rincian total biaya

		masukkan data diri peserta, kemudian klik daftar.	dan mengarahkan ke menu riwayat pemesanan.
MU6	Mengirimkan Bukti Pembayaran	Klik profil, klik Riwayat pemesanan, klik pembayaran, masukan bukti pembayaran dan kirim.	Sistem akan mengirimkan notifikasi apabila pembayaran telah dikonfirmasi.
MU7	Melihat Riwayat Transaksi	Klik profil, klik Riwayat pemesanan	Menampilkan halaman Riwayat transaksi.
MU8	Melihat detail paket yang terdaftar	Klik profil, Riwayat pemesanan, klik detail	Menampilkan rincian pemesanan berupa daftar jemaah, agenda, dan transaksi.

Skenario pengujian pada Tabel 3.17 difokuskan pada fitur-fitur utama yang ditujukan untuk konsumen. Pada tahapan implementasi modul konsumen memiliki situs web yang terpisah dari karyawan. Hal ini ditujukan untuk mempermudah dalam penanganan apabila terdapat kesalahan yang terjadi.

Tahapan implementasi pada modul ini dilakukan dalam 2 tahapan yaitu:

#### 1. Implementasi *Back-End*

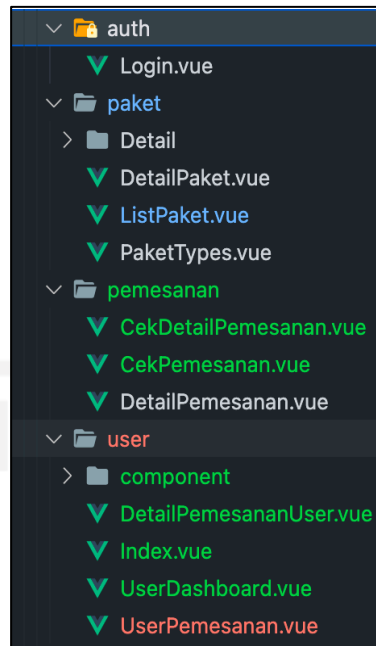
Modul User memiliki banyak aktivitas yang dapat dilakukan oleh konsumen atau Jemaah. Pembuatan API untuk modul user seperti pada Tabel 3.18.

Tabel 3.18 Dokumentasi API User.

No	Method	Path	Deskripsi
1	GET	/paket/	Untuk mengambil list paket.
2	GET	/paket/{kode paket}	Untuk mengambil detail paket.
3	POST	/book/	Untuk melakukan pemesanan baru.
4	GET	/book/	Untuk melihat daftar paket yang dipesan.

#### 2. Implementasi *Front-End*

Implementasi *front-end* pada modul user memiliki beberapa komponen tampilan antara lain: *auth*, pemesanan, paket, dan dashboard untuk *user*. Adapun untuk komponen yang dibuat pada modul *user* seperti pada Gambar 3.14.



Gambar 3.13 Komponen Modul User

```

*/ Potongan skrip model User */
postPemesanan({ dispatch, commit, state }, payload) {
  return new Promise((resolve) => {
    $axios
      .post("/book", payload)
      .then((response) => {
        if (response.data.meta.status) {
          commit("ASSIGN_DATA", response.data.data);
        } else {
          commit("SET_ERRORS", error.response.data.meta.message),
            { root: true };
        }
      })
      .resolve(response.data.data.pemesanan.code);
    })
    .catch((error) => {
      if (!error.response.meta.status) {
        commit("SET_ERRORS", error.response.data.meta.message),
          {
            root: true,
          };
      }
    });
  });
},
}*/ Potongan skrip model User */

```

Gambar 3.14 Potongan Kode Modul User

### *Pair Programming*

Pengembangan modul user dilakukan dengan menggunakan API pada *back-end* yang sama dengan modul admin namun memisahkan tampilan antarmuka dengan membuat *Vue.js* yang terpisah. Pada pengembangan modul user, terdapat perubahan dalam antarmuka yang dibuat oleh

desainer antarmuka sehingga koordinasi dalam setiap perubahan sangat penting dilakukan agar tidak terjadi perbedaan implementasi yang dikembangkan.

### Pengujian

Hasil dari pengujian yang dilakukan dalam modul user dapat dilihat pada Tabel 3.19.

Tabel 3.19 Hasil Pengujian Modul User

No	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Berhasil login dan diarahkan untuk ke beranda.	Sistem mengarahkan ke beranda dengan merubah status login dan menghilangkan tombol login pada menu bar	Berhasil
2	Berhasil mendaftarkan diri dan diarahkan ke beranda dengan status login.	Sistem mengarahkan ke halaman beranda dengan memunculkan notifikasi "Pendaftaran Berhasil" dan merubah status login.	Berhasil
3	Menampilkan daftar paket yang tersedia.	Sistem menampilkan daftar paket tour yang tersedia	Berhasil
4	Menampilkan detail paket yang tersedia.	Sistem menampilkan rincian paket berupa nama paket, foto, deskripsi paket, dan info harga paket	Berhasil
5	Berhasil melakukan pendaftaran, sistem akan memberikan rincian total biaya dan mengarahkan ke menu riwayat pemesanan.	Sistem mengarahkan ke halaman detail pemesanan yang berisi data pemesan, rincian pembayaran, dan keterangan Jemaah.	Berhasil
6	Sistem akan mengirimkan notifikasi apabila pembayaran telah dikonfirmasi.	Sistem mengirimkan email ke pemesan ketika pembayaran dikonfirmasi dalam bentuk invoice.	Berhasil
7	Menampilkan halaman Riwayat transaksi.	Sistem menampilkan daftar transaksi yang telah dilakukan oleh pemesan.	Berhasil
8	Menampilkan rincian pemesanan berupa daftar jemaah, agenda, dan transaksi.	Sistem menampilkan detail pemesanan yang dilakukan berupa data Jemaah, agenda tour, dan pembayaran yang telah dilakukan.	Berhasil

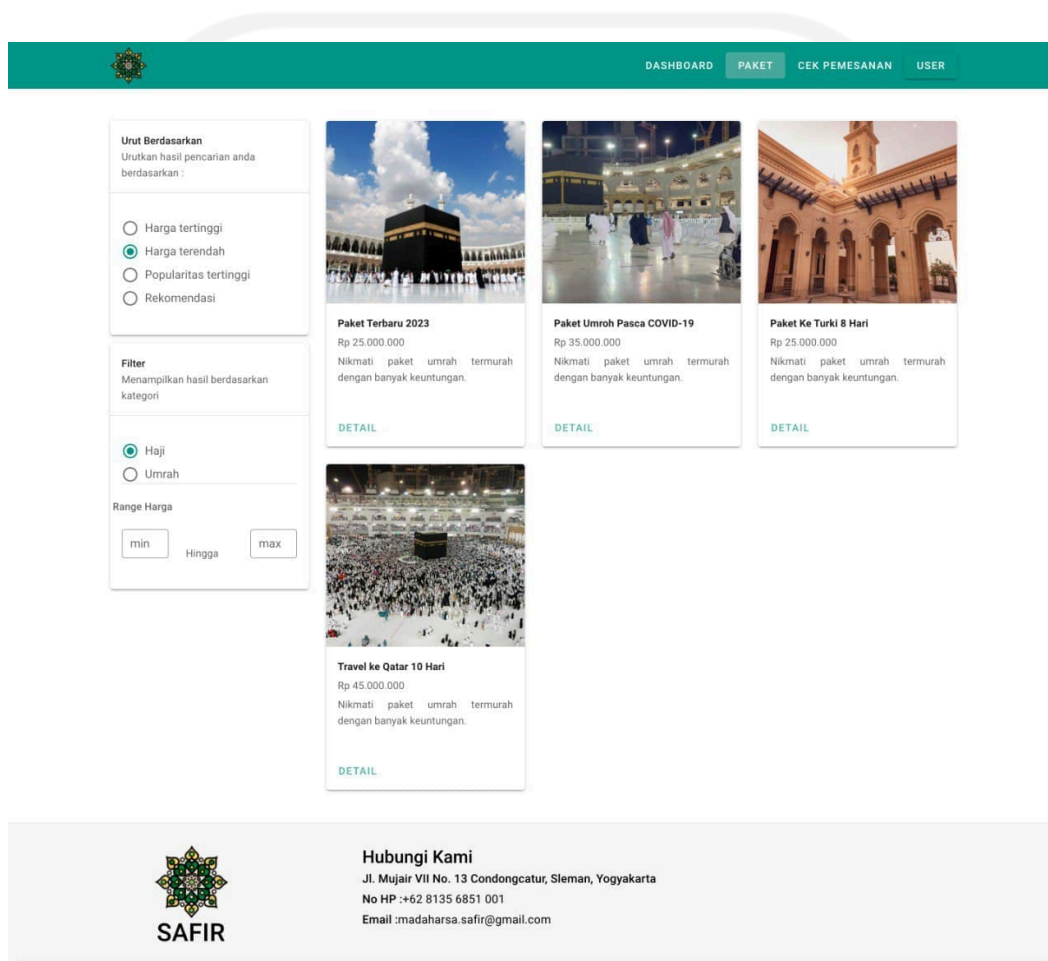
### 3.5 Aspek Dasar Implementasi Extreme Programming pada Safir

Dalam *Extreme Programming* memiliki aspek dasar yang digunakan agar penggunaan metode pengembangan ini dapat terlaksana dengan lebih baik untuk menyelesaikan pengembangan tepat waktu.



### 3.5.1 Simple Design

Dalam pengembangan sistem informasi manajemen travel agen, Safir menggunakan kerangka kerja antar muka yang telah disiapkan oleh *vueify*. Pada tampilan antarmuka, warna yang digunakan dominan putih dan hijau. Dengan rancangan antarmuka yang dibuat meminimalisir banyaknya objek gambar yang digunakan dan lebih mementingkan informasi yang diberikan seperti pada Gambar 3.16.



Gambar 3.15 Tampilan Antar Muka *List Paket*

### 3.5.2 Refactoring

Dalam pengembangan menggunakan metode *Extreme Programming*, *refactoring* menjadi salah satu aspek yang harus diperhatikan. Pada penggunaan aspek ini, pengembang dituntut untuk dapat terus mengembangkan bentuk skrip kode yang simpel, dan mudah dipahami oleh tim pengembang lainnya. Salah satu bentuk *refactoring* yang digunakan yaitu merubah fungsi pada *controller Laravel* seperti pada Gambar 3.17 menjadi seperti Gambar 3.18.

```
public function storeBefore(Request $request)
{
    DB::beginTransaction();
```

```

$book = $request['book'];
$totalPrice = 0;
foreach ($book as $b) {
    $tipe = Tipe::findOrFail($b['id_paket']);
    if ($tipe) $totalPrice += $tipe->harga;
}
$arrJemaah = [];
try {
    $transaksi = $this->transaksi;
    $transaksi->tanggal = Carbon::create(today()->format('Y-m-d'));
    $transaksi->total_harga = $totalPrice;
    $transaksi->sudah_dibayar = 0;
    $transaksi->belum_dibayar = $totalPrice;
    $transaksi->save();

    $pemesanan = $this->pemesanan;
    $pemesanan->nama = $request['nama'];
    $pemesanan->alamat = $request['alamat'];
    $pemesanan->email = $request['email'];
    $pemesanan->no_hp = $request['no_hp'];
    $pemesanan->catatan = $request['catatan'];
    if (isset($request['referral'])) $pemesanan->referral =
$request['referral'];
    $pemesanan->id_transaksi = $transaksi->id;
    $pemesanan->code = $this->getCodeForPemesanan();
    $pemesanan->user_id = Auth::user()->id;
    $pemesanan->save();

    foreach ($request['book'] as $book) {
        $paket = $this->paket->find($book['id_paket']);
        $paket->kuota = $paket->kuota - 1;
        $paket->save();

        $jemaah = new $this->jemaah;
        $jemaah->nama = $book['nama'];
        $jemaah->jenis_kelamin = $book['jenis_kelamin'];
        $jemaah->usia = $book['usia'];
        $jemaah->id_pemesanan = $pemesanan->id;
        $jemaah->id_paket = $paket->id;
        $jemaah->save();
        array_push($arrJemaah, ["jemaah" => $jemaah, "paket" =>
$paket]);
    }

    DB::commit();
    $data = [
        "transaksi" => $transaksi,
        "pemesanan" => $pemesanan,
        "book" => $arrJemaah
    ];
    return $this->setResult($data)->setFail(false);
} catch (\Exception $e) {
    Log::error($e->getMessage());
    DB::rollBack();
    return $this->setMessage($e->getMessage())->setFail(true);
}
}

```

Gambar 3.16 Potongan Fungsi Sebelum *Refactoring*.

```
public function store(PostBookPemesanan $request)
```

```

{
    $result = $this->pemesananService->store($request->all());
    if ($result->isFail()) {
        return $this->fail($result->getMessage());
    }

    return $this->successWithData(
        trans('message.store-data') . ' ' . trans('message.success'),
        $result->getResult()
    );
}

```

Gambar 3.17 Potongan Fungsi Setelah *Refactoring*.

Potongan fungsi yang berhubungan dengan pertukaran data pada *database* dimasukan menjad satu servis yaitu *PemesananService*. Sehingga pada *controller* menjadi lebih sederhana dan terstruktur.

### 3.5.3 Testing

Pada pengembangan yang mengimplementasikan metode *Extreme Programming*, pengujian atau *testing* dilakukan disetiap iterasi pengembangan selesai dilakukan. Pada metode ini, pengujian dilakukan sesuai dengan skema yang dibuat sebelum sistem dibuat. Pengujian dilakukan agar memastikan sistem yang dikembangkan sesuai dengan keinginan klien maupun pengguna atau kostumer.

### 3.5.4 Pair Programming

*Pair Programming* diterapkan dalam pengembangan sistem Safir. Penggunaan *pair programming* yang seharusnya dilakukan berpasangan, pada pengembangan ini *pair programming* dilakukan oleh satu orang *programmer* dan dengan melakukan diskusi pada komunitas maupun forum untuk memudahkan dalam pengembangan. Serta membuat dokumentasi dalam kode yang dikembangkan agar memudahkan ketika tim dalam bentuk yang lebih besar.

### 3.5.5 Coding Standards

Untuk menunjang pemahaman tim pengembang ke depan, diperlukan standarisasi sturktur kode yang digunakan. Tujuannya adalah untuk memberikan pemahaman yang sama dalam pengembangan, dan meminimalisir terjadinya kesalahan dalam pengembangan sistem. Pada pengembangan ini, *coding standart* yang diterapkan merupakan metode yang umum digunakan oleh pengembang. Memisahkan pertukaran data dengan *business logic*. Pengambilan data dilakukan melalui *Repository* dan *business logic* melalui *Service*. Adapun contoh penggunaannya seperti pada Gambar 3.19 dan Gambar 3.20.

```

public function store(array $data)
{
    $start = Carbon::parse($data['start_date']);
    $end = Carbon::parse($data['end_date']);
    $diff = $start->diffInDays($end);
}

```

```

DB::beginTransaction();
try {
    $paket = $this->paket;
    $paket->kode = $data['code'];
    $paket->nama = $data['name'];
    $paket->musim = $data['season'];
    $paket->jml_hari = $diff;
    $paket->tgl_mulai = $start;
    $paket->tgl_berakhir = $end;
    $paket->id_hotel = $data['hotel_id'];
    $paket->id_maskapai = $data['airlines_id'];
    $paket->status = "process";
    $paket->save();

    DB::commit();
    return $this->setResult($paket)->setFail(false);
} catch (\Exception $e) {
    Log::info($e->getMessage());
    DB::rollBack();
    return $this->setMessage($e->getMessage())->setFail(true);
}
}

```

Gambar 3.18 Contoh Potongan Kode *OperationalService.php*.

```

public function show($id)
{
    return $this->paket->with('hotel', 'maskapai', 'tipe', 'photo')
        ->where('id', $id)->first();
}

```

Gambar 3.19 Contoh Potongan Kode *OperationalRepository.php*.

Penggunaan yang berhubungan dengan memodifikasi data, penambahan data, maupun penghapusan data, dipusatkan pada *Service*. Sedangkan untuk penggunaan dengan tujuan penarikan data, detail data, maupun pencarian data, dipusatkan pada *Repository*. *Service* dan *Repository* akan dikelola melalui *controller* yang dapat diakses dari *endpoint* yang dibuat.

## BAB IV

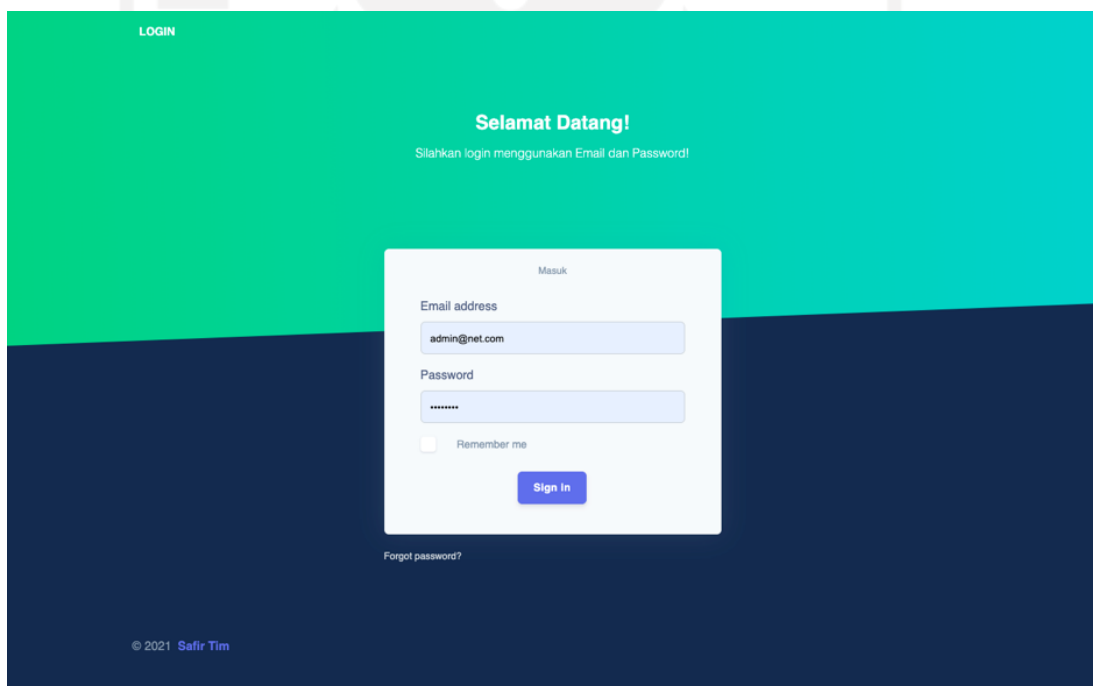
### HASIL DAN PEMBAHASAN

#### 4.1 Hasil

Berdasarkan proses pengembangan sistem informasi manajemen Safir menggunakan *Laravel framework* dengan mengimplementasikan metode pengembangan *Extreme Programming* yang telah dilakukan, maka didapatkan hasil halaman *website* untuk admin, halaman *website* untuk *customer* atau Jemaah serta API yang telah dibuat.

##### 4.1.1 Halaman Website Dashboard Admin

Halaman Admin ditujukan untuk karyawan ataupun staff dari pihak travel agen untuk mengatur tour maupun paket yang dikelola. Halaman ini berisikan modul-modul yang ditujukan untuk dapat mempermudah dalam pendataan Jemaah, pendataan paket, transaksi, hingga pembayaran kustomer atau Jemaah.



Gambar 4.1 Tampilan Halaman Login

Tampilan awal dari halaman admin adalah Login, seperti terlihat pada Gambar 4.1, Login pada halaman admin menggunakan email dan password.

**SAFIR DASHBOARD**

Search  Admin Operasional

**TOTAL JEMAAH AKTIF**: 5

**TOTAL PAKET AKTIF**: 10

**TOTAL PEMBAYARAN**: Rp285.700.000

**BELUM VERIFIKASI**: 5

**Pembayaran Pending** [See all](#)

NAMA JEMAAH	PAKET	STATUS	NOMINAL
BAMBANG	PKT0512	belum_terkonfirmasi	Rp. 5.000.000
ALI LALI	TRK001	belum_terkonfirmasi	Rp. 10.000.000
DIKA	PKT0512	belum_terkonfirmasi	Rp. 25.000.000
DIKA	PKT1234	belum_terkonfirmasi	Rp. 32.000.000
AJI	TRK001	belum_terkonfirmasi	Rp. 3.000.000

© 2021 Safir Tim

Gambar 4.2 Tampilan Halaman Dashboard

Beranda awal laman web admin bertujuan untuk memberikan informasi yang dihasilkan dari data yang tersedia. Admin dapat melakukan manajemen salah satunya manajemen paket tour dan travel. Seperti yang terlihat pada Gambar 4.3 hingga Gambar 4.6.

**SAFIR PAKET**

Search  Admin Operasional

**Daftar Paket**

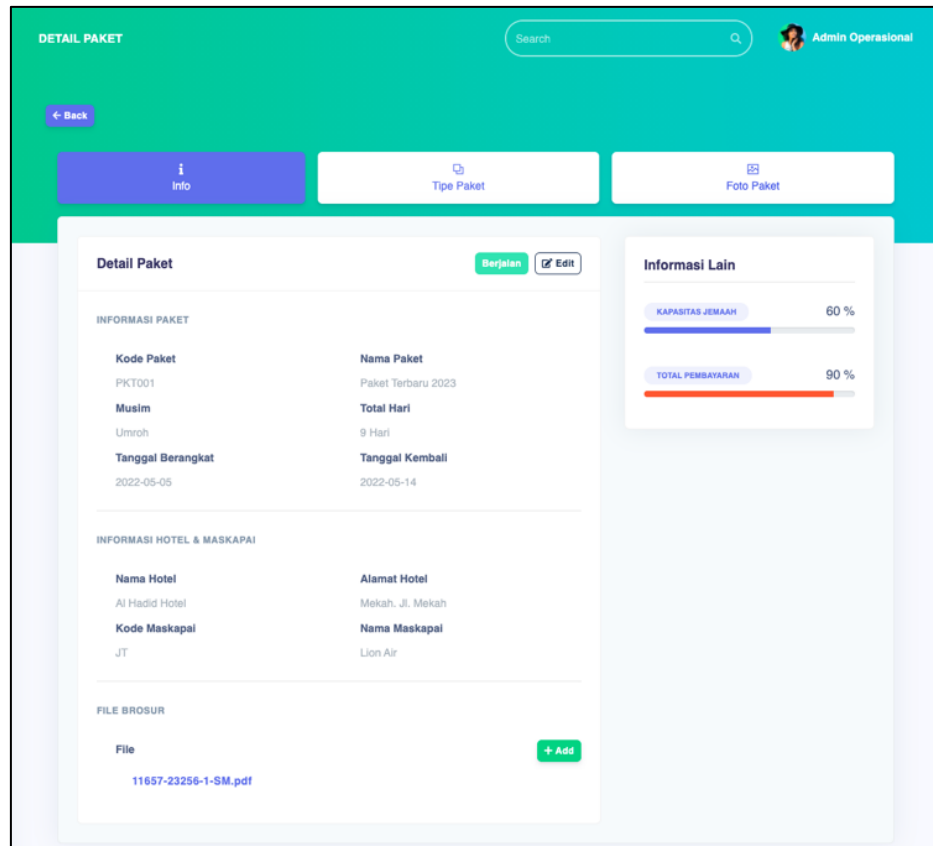
Search  [+ Add](#)

KODE	NAMA PAKET	MUSIM	STATUS	TANGGAL
PKT001	Paket Terbaru 2023	Umroh	process	2022-05-05   2022-05-14
PKT123	Paket Turki	Haji	cancelled	2021-10-21   2021-10-21
PKT423	Paket Umroh Pasca COVID-19	Haji	done	2021-08-15   2021-08-25
TURKI200	Paket Ke Turki 8 Hari	Haji	process	2022-07-21   2022-07-29
QTR001	Travel ke Qatar 10 Hari	Travel	process	2022-08-24   2022-08-30

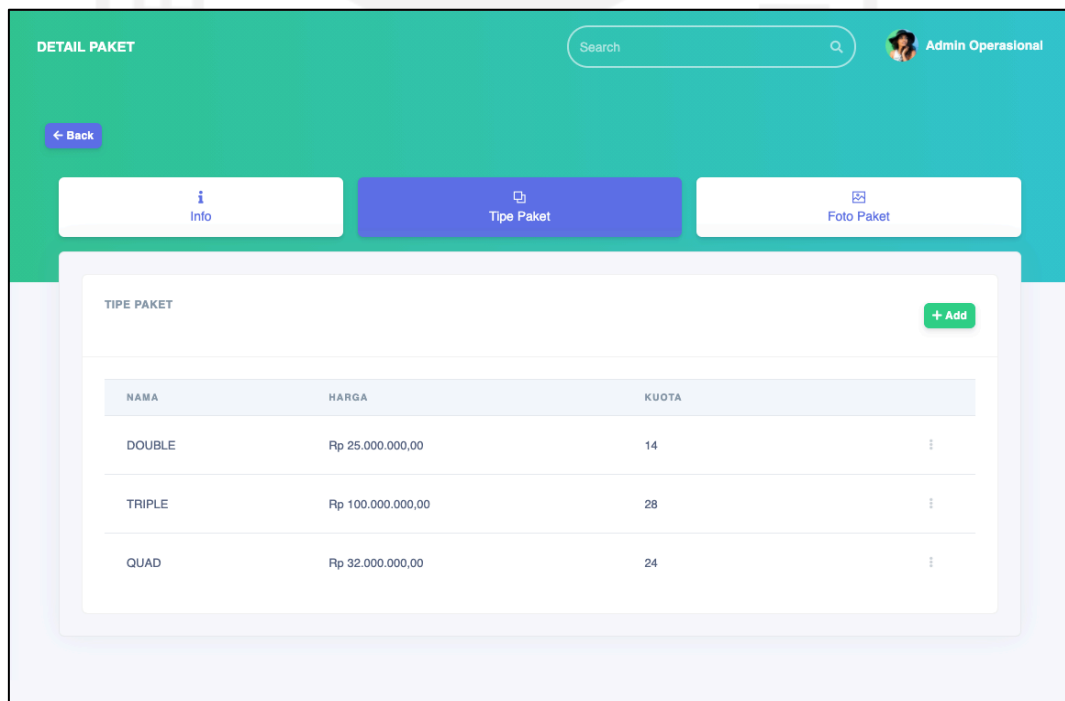
1 hingga 5 dari 5 total data

© 2021 Safir Tim

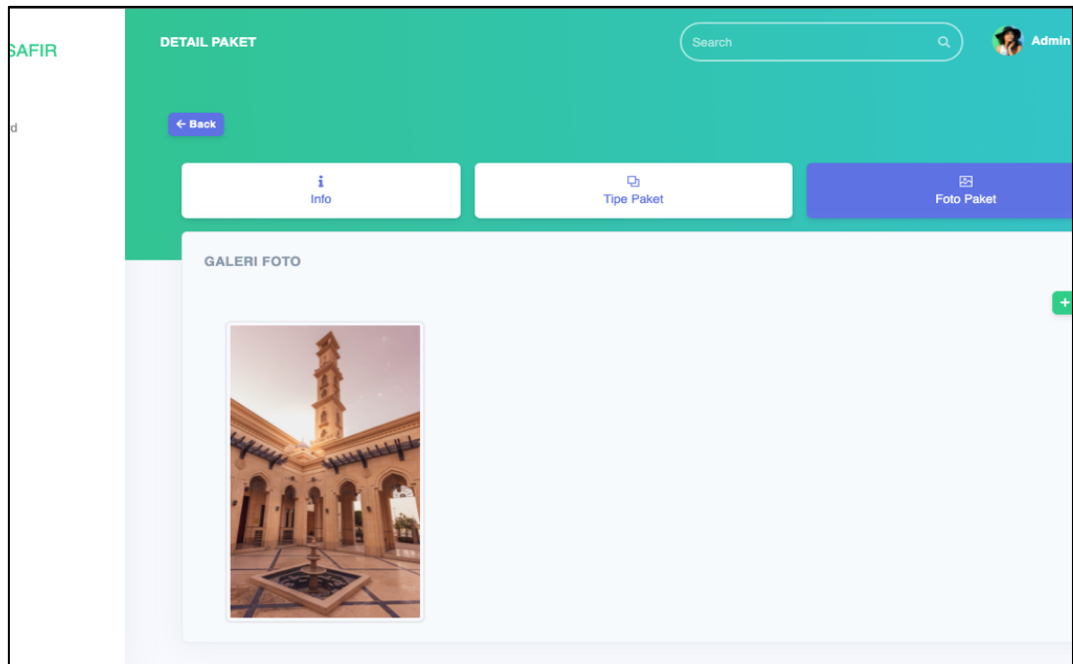
Gambar 4.3 Tampilan Halaman Paket



Gambar 4.4 Tampilan Halaman Detail Paket

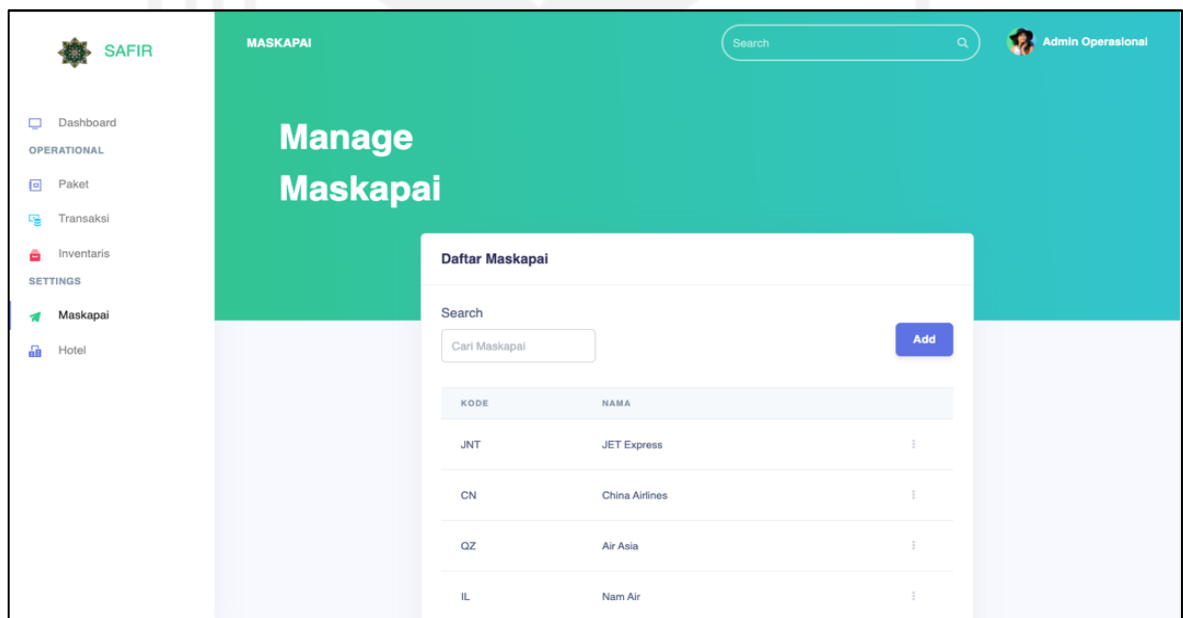


Gambar 4.5 Tampilan Halaman Detail Tipe Paket



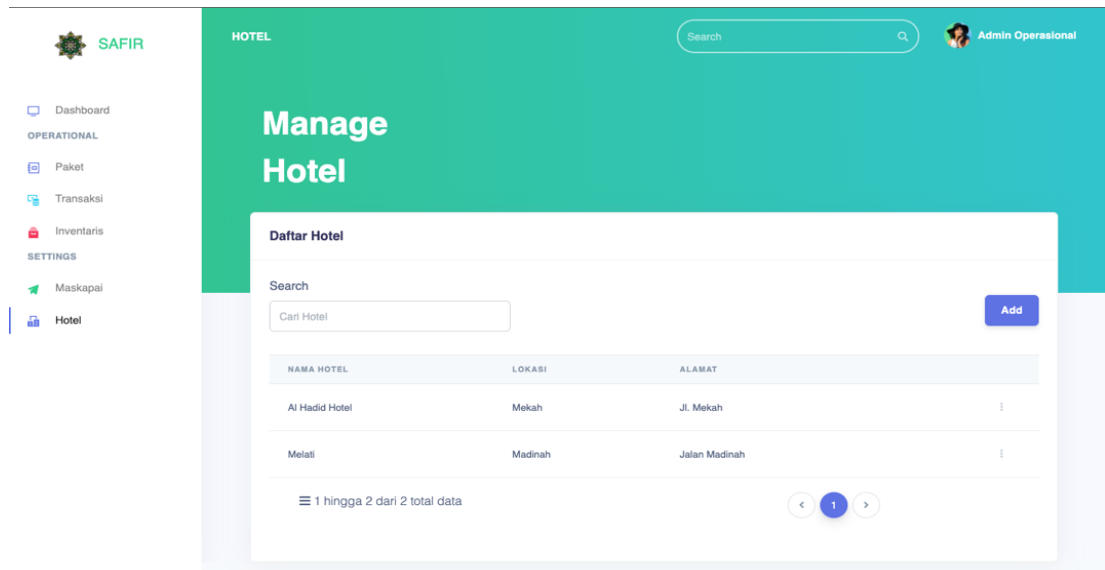
Gambar 4.6 Tampilan Halaman Detail Galeri Paket

Admin dapat mengatur data-data maskapai dan hotel yang akan digunakan sebagai maskapai dan hotel yang dipakai pada saat perjalanan. Seperti terlihat pada Gambar 4.7 dan Gambar 4.8.



Gambar 4.7 Tampilan Halaman Maskapai

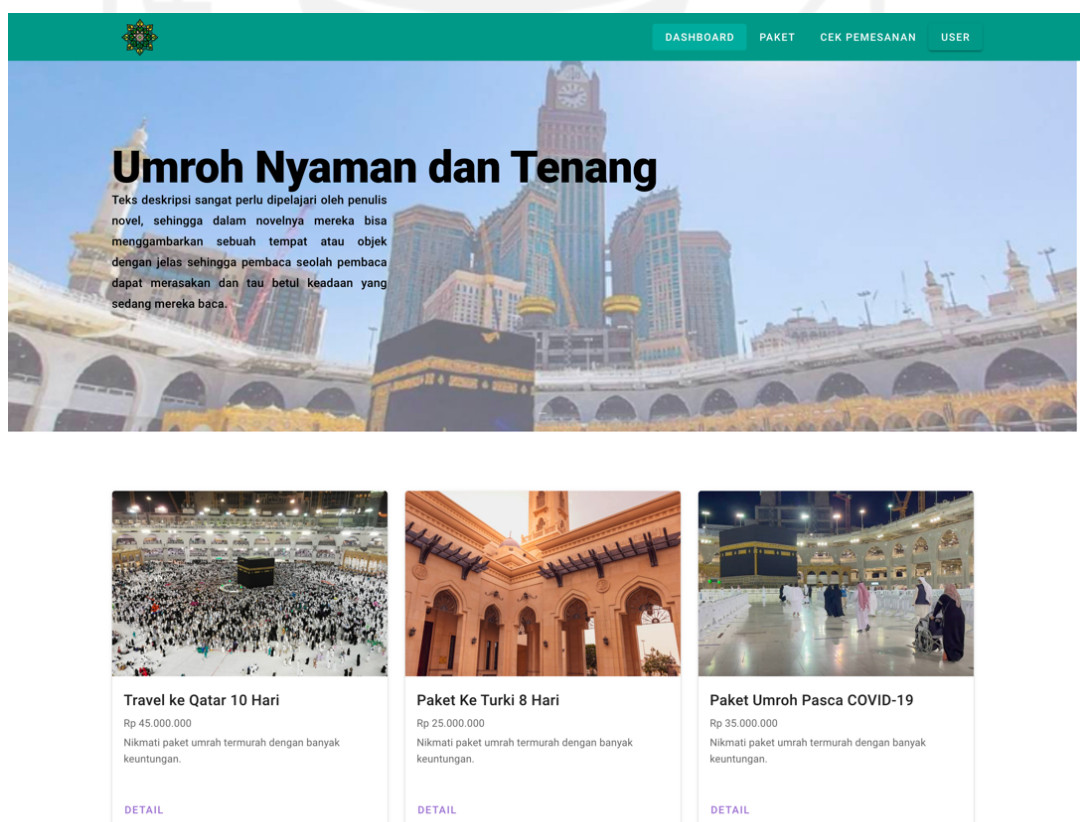


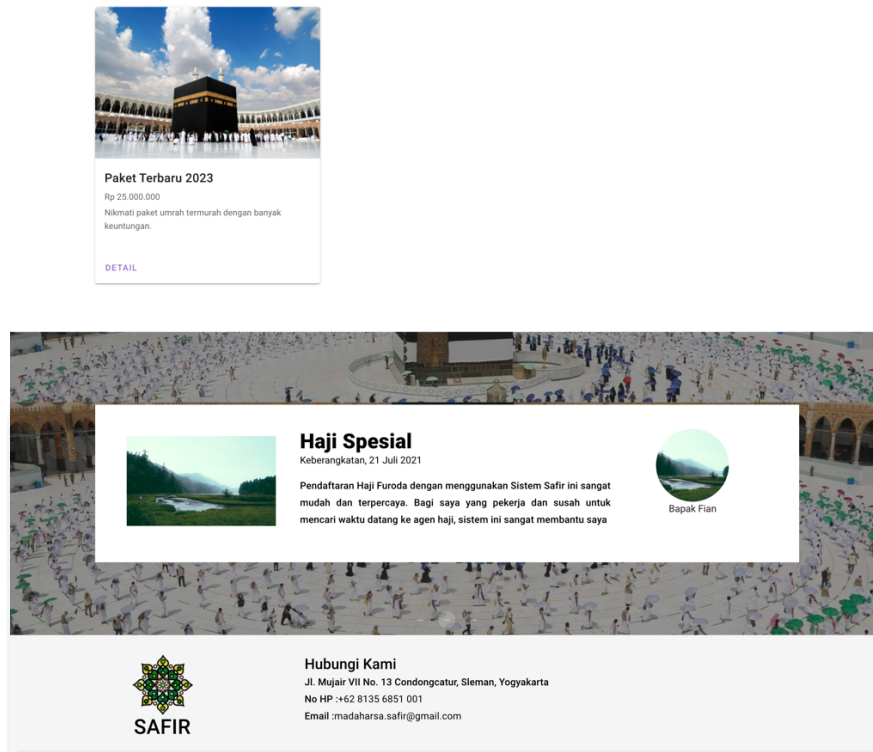


Gambar 4.8 Tampilan Halaman Hotel

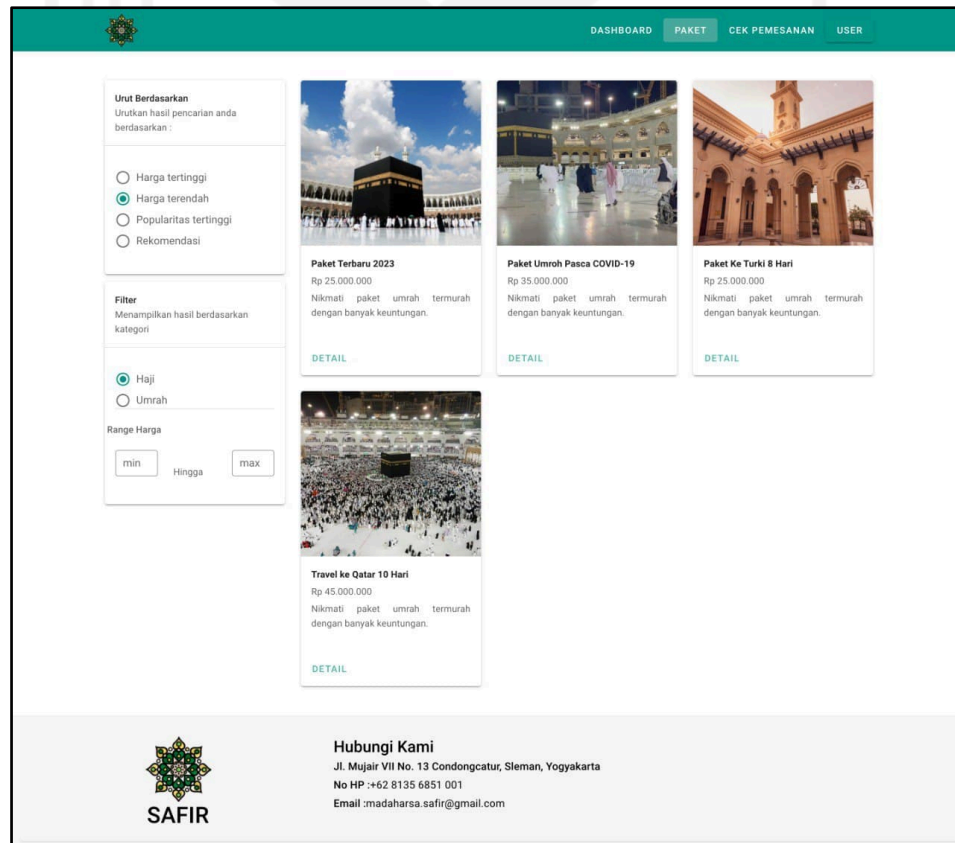
#### 4.1.2 Halaman Website Dashboard Customer

Halaman *website customer* ditujukan untuk transaksi yang dilakukan oleh *customer*. Dimulai dari lihat daftar paket, pemesanan, hingga kelengkapan data Jemaah.





Gambar 4.9 Tampilan Halaman Depan Kustomer



Gambar 4.10 Tampilan Halaman List Paket

**Paket Umroh Pasca COVID-19**

Keberangkatan : 2021-08-25

Paket ini merupakan paket yang ekonomis, fasilitas yang disediakan sangat terpenuhi meskipun dengan harga yang sangat terjangkau. Tour guide yang sangat ramah dan juga bimbingan yang diberikan akan sangat membantu jamaah yang akan berangkat umrah. Pada paket ini jamaah akan mendapatkan tujuan ke berbagai tempat seperti Makam Nabi Muhammad SAW, Goa Hiro, DII.

**Fasilitas**

- Hotel dekat dengan Masjidil Haram
- Pesawat menggunakan Flynas Airlines
- 1 minggu di Arab Saudi
- 4 hari di Makkah, 3 hari di Madinah
- Transportasi dari rumah ke bandara sudah disediakan
- Kartu identitas disediakan

**Hubungi Kami**  
 Jl. Mujair VII No. 13 Condongcatur, Sleman, Yogyakarta  
 No HP : +62 8135 6851 001  
 Email : madaharsa.safir@gmail.com

Gambar 4.11 Tampilan Halaman Detail Paket

**Paket Umroh Pasca COVID-19**

Keberangkatan : 2021-08-25

Sisa Kuota 48 pax | Batas Pendaftaran 10 Hari Lagi

Paket Paket Umroh Pasca COVID-19 keberangkatan Selama 10 hari. Mulai dari Rp. 28.000.000

**Bantuan**

Silahkan masukkan jumlah orang sesuai dengan tipe paket yang dipilih ; contoh 2 Orang Dewasa di tipe superior, makan pada bagian SUPERIOR inputkan 2 dewasa.

Gambar 4.12 Tampilan Halaman Pemesanan Paket

**Nama Pemesan : Dika**

Nomor HP : 081234567890  
 Email : dika@mail.com

Tipe Paket	Nominal
1 (DOUBLE)	Rp 35.000.000
<b>TOTAL HARGA</b>	<b>Rp 35.000.000</b>

**Uang Muka**  
 Rp 3.500.000

Untuk dapat melanjutkan proses selanjutnya, silahkan lakukan pembayaran uang DP terlebih dahulu!

**Paket Umroh Pasca COVID-19**

Keberangkatan : 2021-08-25

Sisa Kuota 48 pax | Batas Pendaftaran 10 Hari Lagi

Paket Paket Umroh Pasca COVID-19 keberangkatan 2021-08-25 Selama 10 hari. Mulai dari Rp. 28.000.000

**Transfer Bank**

**BANK SYARIAH INDONESIA**  
 Atas Nama :  
 PT APLIKASI HAJI DAN UMRAH  
 No. Rekening : 1234567890

**BANK MANDIRI**  
 Atas Nama :  
 PT APLIKASI HAJI DAN UMRAH  
 No. Rekening : 1234567890

**Pembatalan dan Pengembalian**

Jika pembatalan keberangkatan dilakukan 31 - 100 hari sebelum jadwal keberangkatan, maka pihak Ithram Asia akan mengembalikan dana jamaah setelah dipotong sebesar 5%.

Jika pembatalan keberangkatan dilakukan 15 - 30 hari sebelum jadwal keberangkatan, maka pihak Ithram Asia akan mengembalikan dana jamaah setelah dipotong sebesar 15%.

Jika pembatalan keberangkatan dilakukan 8 - 14 hari sebelum jadwal keberangkatan, maka pihak Ithram Asia akan mengembalikan dana jamaah setelah dipotong sebesar 30%.

Jika pembatalan keberangkatan dilakukan 4 - 7 hari sebelum jadwal keberangkatan, maka pihak Ithram Asia akan mengembalikan dana jamaah setelah dipotong sebesar 75%.

Gambar 4.13 Tampilan Halaman Detail Pemesanan



*Response:*

```
{
  "meta": {
    "code": 200,
    "message": "data success",
    "status": true
  },
  "data": {
    "current_page": 1,
    "data": [
      {
        "id": 4,
        "name": "Al Hadid Hotel",
        "address": "Jl. Mekah",
        "location": "Mekah"
      },
      {
        "id": 3,
        "name": "Melati",
        "address": "Jalan Madinah",
        "location": "Madinah"
      }
    ],
    "first_page_url": "http://safir-be.test/api/admin/hotel?page=1",
    "from": 1,
    "last_page": 1,
    "last_page_url": "http://safir-be.test/api/admin/hotel?page=1",
    "links": [
      {
        "url": null,
        "label": "&laquo; Previous",
        "active": false
      },
      {
        "url": "http://safir-be.test/api/admin/hotel?page=1",
        "label": "1",
        "active": true
      },
      {
        "url": null,
        "label": "Next &raquo;",
        "active": false
      }
    ],
    "next_page_url": null,
    "path": "http://safir-be.test/api/admin/hotel",
    "per_page": 10,
    "prev_page_url": null,
    "to": 2,
    "total": 2
  }
}
```

Gambar 4.16 Response List Hotel.

- **Create Hotel**

API untuk menambahkan data hotel baru.

Tabel 4.3 API POST Hotel

No	Parameter	Deskripsi
1	<i>endpoint</i>	{base_url}/admin/paket
2	<i>Method</i>	POST
3	Header	Accept ⇒ application/json Authorization ⇒ Bearer *token

*Request Body:*

```
{
  "code" : "PKT223",
  "name" : "Paket Terbaru 2023",
  "season" : "Haji",
  "start_date" : "2022-05-05",
  "end_date" : "2022-05-14",
  "hotel_id" : 4,
  "airlines_id" : 4
}
```

Gambar 4.17 Request Body Post Paket.

*Response:*

```
{
  "meta": {
    "code": 200,
    "message": "add data success",
    "status": true
  },
  "data": {
    "uid": 12,
    "code": "PKT223",
    "name": "Paket Terbaru 2023",
    "season": "Haji",
    "status": "process",
    "total_days": 9,
    "start_date": "2022-05-05T00:00:00.000000Z",
    "end_date": "2022-05-14T00:00:00.000000Z",
    "hotel_id": 4,
    "airlines_id": 4,
    "hotel": {
      "id": 4,
      "name": "Al Hadid Hotel",
      "alamat": "Jl. Mekah",
      "lokasi": "Mekah"
    },
    "airlines": {
      "id": 4,
      "kode_maskapai": "JT",
      "nama": "Lion Air"
    },
    "types": [],
    "photos": [],
    "hotels": {
      "id": 4,
```

```

        "name": "Al Hadid Hotel",
        "alamat": "Jl. Mekah",
        "lokasi": "Mekah"
    },
    "desc": "Paket ini merupakan paket yang ekonomis, fasilitas yang
disediakan sangat terpenuhi meskipun dengan harga yang sangat terjangkau. Tour
guide yang sangat ramah dan juga bimbingan yang diberikan akan sangat membantu
jamaah yang akan berangkat umrah. Pada paket ini jamaah akan mendapatkan tujuan
ke berbagai tempat seperti Makam Nabi Muhammad SAW, Goa Hiro, Dll."
    }
}

```

Gambar 4.18 Response Post Paket.

### API untuk User

Berikut beberapa API yang telah dibuat pada modul yang digunakan oleh *User*:

- **Auth**

*Auth* ditujukan untuk mendapatkan *token* yang akan digunakan pada setiap *request* pada *endpoint* yang digunakan.

Tabel 4.4 API Auth User

No	Parameter	Deskripsi
1	<i>endpoint</i>	{base url}/user/login
2	<i>Method</i>	POST
3	Header	Accept ⇒ application/json

#### Request Body:

```

{
  "email" : "user@net.com",
  "password" : "Admin123"
}

```

Gambar 4.19 Request Body Auth.

#### Response:

```

{
  "meta": {
    "code": 200,
    "message": "Success Login.",
    "status": true
  },
  "data": {
    "id": 1,
    "name": "user",
    "email": "user@net.com",
    "email_verified_at": "2021-07-15T10:41:36.000000Z",
    "is_admin": 1,
    "created_at": "2021-07-15T10:41:36.000000Z",
    "updated_at": "2021-07-15T10:41:36.000000Z",
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1b290eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1b290eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9"
  }
}

```

Gambar 4.20 Response Auth.

- **List Paket**

API untuk memberikan data *list* paket yang tersedia.

Tabel 4.5 API Get List Paket

No	Parameter	Deskripsi
1	<i>endpoint</i>	{base_url}/user/paket
2	<i>Method</i>	GET
3	Header	Accept ⇒ application/json

*Response:*

```
{
  "meta": {
    "code": 200,
    "message": "data",
    "status": true
  },
  "data": [
    {
      "id": 7,
      "kode": "PKT001",
      "nama": "Paket Terbaru 2023",
      "musim": "Umroh",
      "jml_hari": 9,
      "tgl_mulai": "2022-05-05",
      "tgl_berakhir": "2022-05-14",
      "id_hotel": 4,
      "id_maskapai": 4,
      "created_at": "2021-08-04T10:09:18.000000Z",
      "updated_at": "2021-08-30T08:20:29.000000Z",
      "status": "process",
      "photo": [
        {
          "id": 33,
          "id_paket": 7,
          "name": "e2f1919632030c0db045e970b21d1530.jpg",
          "url":
            "https://www.adira.co.id/img/frontend/news/thumb_Inilah5PerbedaanHajidanUmrah
            BesertaBiayanya.jpg",
          "path":
            "public/images/PKT001/lZrTHOEFhBCCaTH8jpMqx8NVXxxkF1WIGP9aAGfh.jpg",
          "created_at": "2021-08-30T13:27:04.000000Z",
          "updated_at": "2021-08-30T13:27:04.000000Z",
          "link": "http://safir-
            be.test//storage/images/PKT001/lZrTHOEFhBCCaTH8jpMqx8NVXxxkF1WIGP9aAGfh.jpg"
        },
        {
          "id": 35,
          "id_paket": 7,
          "name": "5451_ho_00_p_1024x768.jpg",
          "url": "http://safir-
            be.test/storage/images/PKT001/hSdMzv2NrnKnCJw5yi6SqC8VdzH3SEKyarNVVJgY.jpg",
          "path":
            "public/images/PKT001/hSdMzv2NrnKnCJw5yi6SqC8VdzH3SEKyarNVVJgY.jpg",
          "created_at": "2021-08-30T13:28:08.000000Z",
          "updated_at": "2021-08-30T13:28:08.000000Z",
        }
      ]
    }
  ]
}
```



```

        "link": "http://safir-
be.test//storage/images/PKT001/hSdMzv2NrnKnCJw5yi6SqC8VdzH3SEKyarNVVJgY.jpg"
    },
    "tipe": [
        {
            "id": 6,
            "nama": "DOUBLE",
            "harga": 25000000,
            "kuota": 14,
            "id_paket": 7,
            "created_at": "2021-08-05T21:02:36.000000Z",
            "updated_at": "2021-10-24T02:25:37.000000Z"
        },
        {
            "id": 16,
            "nama": "TRIPLE",
            "harga": 100000000,
            "kuota": 28,
            "id_paket": 7,
            "created_at": "2021-08-26T09:30:17.000000Z",
            "updated_at": "2021-10-24T02:25:43.000000Z"
        },
        {
            "id": 19,
            "nama": "QUAD",
            "harga": 32000000,
            "kuota": 24,
            "id_paket": 7,
            "created_at": "2021-08-30T16:29:59.000000Z",
            "updated_at": "2021-10-24T02:25:49.000000Z"
        }
    ]
}

```

Gambar 4.21 *Response List Paket.*

## 4.2 Pembahasan

Pengembangan sistem informasi manajemen haji dan umroh dengan *Laravel framework* yang telah menggunakan metode pengembangan *Extreme Programming (XP)* untuk Safir, maka terdapat beberapa kelebihan dan kekurangan selama pengembangan berlangsung.

### 4.2.1 Kelebihan

Keuntungan ataupun kelebihan yang didapat selama pengembangan sistem informasi manajemen haji dan umroh untuk Safir sebagai berikut:

- a. Pengembangan *back-end* menggunakan *Laravel Framework* yang bersifat *open-source* sehingga pada pengembangan dapat dilakukan secara gratis, tidak berbayar, ataupun berlangganan.
- b. *Laravel* merupakan salah satu *framework* yang sangat populer di kalangan pengembang dalam membangun API. Terutama pengembang yang memiliki dasar PHP.

- c. Manajemen proyek yang diberikan oleh *Laravel* sangat mudah digunakan dan terstruktur, seperti pembuatan *Model, Controller, Transformer*, maupun *Routing API*.
- d. *Laravel* memberikan dokumentasi yang baik dan tersusun. Selain itu komunitas PHP Indonesia yang sangat membantu dalam setiap ada kesulitan maupun kendala dalam pengembangan Safir.
- e. Pengembangan menggunakan metode *Extreme Programming* sangat membantu dalam mengatasi adanya perubahan yang cepat berdasarkan kebutuhan klien yang berubah-ubah.
- f. Interaksi dua arah dengan klien sangat berperan penting dalam meminimalisir terjadinya kesalahan dalam pengembangan dan tetap sesuai dengan kebutuhan yang diminta klien.
- g. Implementasi *Extreme Programming* menuntut untuk pengembang dapat melakukan inovasi dan mengembangkan diri untuk menunjang pengembangan sistem semakin baik.
- h. *Refactoring, pair programming*, dan *coding standards* menjadi beberapa aspek yang sangat bagus untuk diimplementasikan dalam pengembangan sistem. Karena aspek-aspek tersebut membuat pengerjaan lebih terstruktur dan memiliki pemahaman yang merata.
- i. Pada pengembangan ini terdapat beberapa *tools* yang sangat membantu dalam efisiensi proses pengembangan sehingga dapat membuat proses pengembangan menjadi lebih singkat.

#### 4.2.2 Kekurangan

Beberapa kekurangan dan hambatan yang timbul dalam proses pengembangan sistem informasi manajemen Safir diantaranya sebagai berikut:

- a. Pengembangan akan sangat terhambat ketika komunikasi dengan klien tidak berjalan dengan baik.
- b. Kurangnya tim pengembang dalam menggunakan metode *Extreme Programming* dapat menjadi kendala ketika perubahan yang dalam jumlah yang banyak dengan skala yang besar terjadi. Karena dalam tim tahapan awal saat ini memiliki satu *programmer*.
- c. Beberapa ahli mengatakan bahwa *Extreme Programming* difokuskan pada kode daripada pada desain. Hal ini mungkin bisa menjadi masalah karena desain yang baik sangat penting untuk aplikasi perangkat lunak serta mempengaruhi nilai jual di pasar perangkat lunak. Selain itu, dalam proyek XP cacatnya dokumentasi tidak selalu baik. Cacatnya dokumentasi merupakan kekurangan yang dapat menyebabkan terjadinya *bug* serupa di masa mendatang.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan pengembangan sistem sebelumnya menggunakan metode *Extreme Programming* (XP) dapat disimpulkan sebagai berikut:

- a. XP merupakan suatu model pengembangan perangkat lunak yang menyederhanakan tahapan-tahapan dalam pengembangan menjadi lebih adaptif dan fleksibel.
- b. Metodologi pengembangan XP ditujukan untuk tim yang berukuran kecil hingga medium.
- c. Metodologi ini memungkinkan perusahaan pengembangan perangkat lunak untuk menghemat biaya dan waktu yang dibutuhkan untuk realisasi proyek.
- d. Umpan balik yang konstan juga merupakan sisi yang kuat. Penting untuk mendengarkan dan membuat perubahan apa pun yang diperlukan tepat waktu.
- e. XP memungkinkan untuk menghemat biaya dan waktu yang dibutuhkan dalam pengembangan karena XP berfokus pada produk akhir yang tepat waktu. Hal ini menjadi kelebihan yang digunakan untuk pengembangan Safir.
- f. Safir merupakan sistem yang berkembang dalam waktu yang singkat. Hal ini merupakan salah satu nilai yang didapat dari XP sehingga dapat memudahkan pengembangan. Komunikasi yang dibangun dengan pengguna menghasilkan berbagai kebutuhan berupa user stories yang digunakan sebagai acuan dalam pengembangan

#### **5.2 Saran**

Berdasarkan hasil pengembangan yang menggunakan metode pengembangan *Extreme Programming* (XP) maka penulis berharap sebagai berikut:

- a. Penggunaan XP perlu menjadi perhatian bahwa metode ini memiliki aspek dasar yang harus diterapkan untuk menunjang keberhasilan dalam menggunakan metode ini.
- b. Interaksi dengan klien akan mempengaruhi kinerja pengerjaan dikarenakan tingginya kompleksitas kebutuhan yang diinginkan sehingga harus menjadi pertimbangan.
- c. Ketika *requirement* yang diberikan sudah stabil dan tidak dinamis, diperlukan adanya evaluasi untuk membahas penggunaan metode ini atau merubah dengan metode yang lebih baik.
- d. Perlu adanya perawatan atau pengembangan terhadap sistem informasi yang dibuat meliputi pengembangan modul, fitur, maupun teknologi yang dipakai agar seiring berkembangnya Safir dan semakin banyaknya pengguna agar dapat ditangani dan sudah disiapkan dengan baik.

## DAFTAR PUSTAKA

- Baniaş, O., Florea, D., Gyalai, R., & Curiac, D. I. (2021). Automated Specification-Based Testing of REST APIs. *Sensors*, 21(16), 5375.
- Chen, X., Ji, Z., Fan, Y., & Zhan, Y. (2017, October). Restful API architecture based on laravel framework. In *Journal of Physics: Conference Series* (Vol. 910, No. 1, p. 012016). IOP Publishing.
- Firman, A., Wowor, H. F., & Najoran, X. (2016). Sistem informasi perpustakaan online berbasis web. *Jurnal Teknik Elektro dan Komputer*, 5(2), 29-36.
- Indrajit, R. E. (2000). *Manajemen sistem informasi dan teknologi informasi*. Jakarta: PT Elex Media Komputindo,
- Nick. (2020, November). *What is REST API?*. Retrieved from blog.iron.io:  
<https://blog.iron.io/what-is-rest-api/>
- Qomplx (2020). *What is vuetify?*. Retrieved from vuetifyjs.com:  
<https://vuetifyjs.com/en/introduction/why-vuetify/>
- Sharo, Z. (2020). *Maintenance Management System* (Doctoral dissertation, University of Zakho).
- So, Preston. (2018). Vue. js. In *Decoupled Drupal in Practice* (pp. 381-397). Apress, Berkeley, CA.
- Sutabri, T. (2012). *Konsep sistem informasi*. Penerbit Andi.
- Widhiartha, P. (2012). *Extreme Programming–Melakukan Pengembangan Perangkat Lunak dengan Lebih Sederhana*.
- S. Roohullah Jan, S. Tauhid Ullah Shah, Z. Ullah Johar, Y. Shah, & F. Khan . (2014). An Innovative Approach to Investigate Various Software Testing Techniques and Strategis. *Int. J. Sci. Res. Sci. Eng. Technol.*
- Widodo, W., & Subekti, M. (2006). *Requirements Management Pada Extreme Programming*. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.
- Yadav, N., Rajpoot, D. S., & Dhakad, S. K. (2019, November). Laravel: a php framework for e-commerce website. In *2019 Fifth International Conference on Image Information Processing (ICIIP)* (pp. 503-508). IEEE.
- Nidhra, S., & Dondeti, J. (2021). *Black Box And White Box Testing Techniques*. A literatur Review

## 2. LAMPIRAN

### a. Basis data Safir

Language: English MySQL » 127.0.0.1 » db\_travel » Select: jemaahs Logout

Adminer 4.8.1

DB: db\_travel

SQL command Import Export Create table

Select data Show structure Alter table New item

Select Search Sort Limit 50 Text length 100 Action Select

```
SELECT * FROM `jemaahs` LIMIT 50 (0.001 s) Edit
```

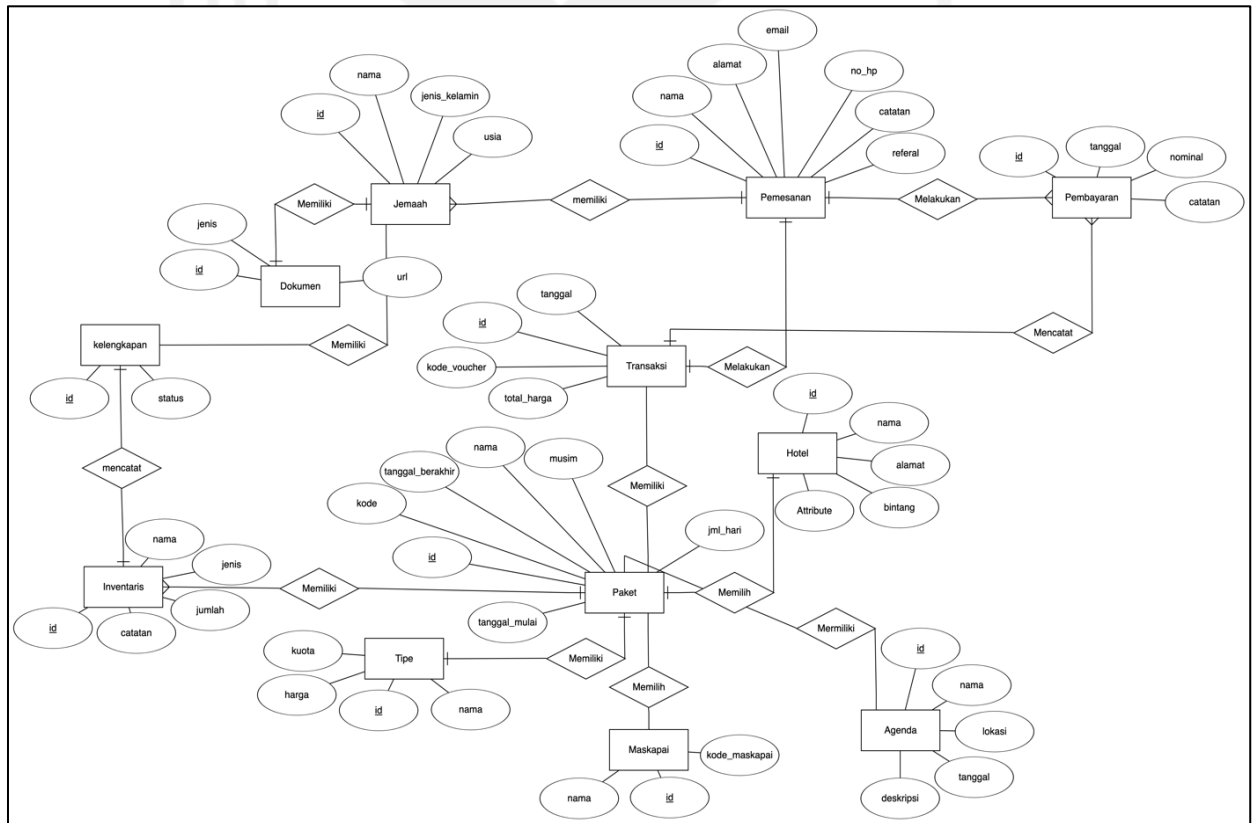
<input type="checkbox"/> Modify	id	nama	jenis_kelamin	usia	id_pemesanan	created_at	updated_at	id_paket
<input type="checkbox"/> edit	1	ALI LALI	P	34	1	2021-09-13 06:40:49	2021-09-13 06:40:49	6
<input type="checkbox"/> edit	2	ALI BABA	P	32	1	2021-09-13 06:40:49	2021-09-13 06:40:49	6
<input type="checkbox"/> edit	3	ALI MANSUR	P	28	1	2021-09-13 06:40:49	2021-09-13 06:40:49	16

Whole result  3 rows  Modify  Selected (0)


Import

- select admins
- select agendas
- select dokuments
- select failed\_jobs
- select file\_pakets
- select foto\_pakets
- select hotels
- select image\_payment
- select inventaris
- select jemaahs
- select kelengkapans
- select maskapais
- select migrations
- select oauth\_access\_tokens
- select oauth\_auth\_codes
- select oauth\_clients
- select oauth\_personal\_access\_client
- select oauth\_refresh\_tokens
- select pakets
- select password\_resets
- select pembayarans
- select pemesanan\_details
- select pemesananans
- select tagihans
- select tipses
- select transaksis
- select users

### b. Entity Relationship Diagram (ERD) Safir



### c. Invoice Pemesanan Safir

	Tanggal Transaksi : 2021-10-24T04:26:07.000000Z			
<b>Safir Indonesia</b>				
Paket	: Paket Umroh Pasca COVID-19			
Nama Pemesan	: Dika			
Email	: dika@mail.com			
No. Telepon	: 081234567890			
Status	: <span style="border: 1px solid black; padding: 2px;">BELUM BAYAR UANG MUKA</span>			
Catatan	:			
<b>Rincian Pemesanan</b>				
#	Tipe Paket	Qty	Satuan	Total
1	DOUBLE	1	Rp 35.000.000	Rp 35.000.000
<b>Rincian Tagihan</b>				
#	Jenis Pembayaran	Status	Nominal	Batas Pembayaran
1	Uang Muka	Belum Bayar	Rp 3.500.000	2021-10-31



#### d. *Dashboard User Safir*

The screenshot displays the 'Safir Dashboard for User' interface. It features a dark sidebar on the left with navigation options: Dashboard, Pemesanan, Settings, and LOGOUT. The main content area is titled 'Dashboard' and contains four summary cards:

Category	Value
Total Pemesanan	5 Pesanan
Belum Terbayarkan	Rp 250.500.000
Sudah Terbayarkan	Rp 16.700.000
Total Tagihan	Rp 267.200.000

