

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Pekerjaan magang selama ini berada pada posisi *React Development* yang berfokus pada pembuatan aplikasi mobile dan website (*Dashboard*). Tugas-tugas magang diberikan melalui website *ClickUp* dengan *deadline* dan tingkat urgensi tertentu. Supervisi divisi akan memberikan tugas dengan melakukan penempatan anggota ke salah satu atau lebih tugas. Tugas-tugas dikerjakan dan dikumpulkan melalui *GitHub* dan komunikasi tim menggunakan *Gmail Chat*. Tugas-tugas yang sudah dikerjakan kemudian dilaporkan melalui supervisi yang nantinya akan di-*review* oleh supervisi dan akan melakukan *Test QA*. Hasil test bergantung dengan apa yang dikerjakan, apabila belum memenuhi kriteria, maka akan dikembalikan ke anggotanya untuk dikerjakan kembali, dan sebaliknya apabila sudah memenuhi, maka akan diimplementasikan ke *production*.

#### 3.2 Tugas yang Dilakukan

Tugas yang dilakukan selama magang mencakup:

1. Pembuatan ulang tampilan pada *dashboard trip* yang mencakup akomodasi, kendaraan, *local guide*, *event*, *ticket and destination*, dan lainnya.
2. Perbaikan *bug* dan penanganan *error* pada berbagai komponen.
3. Pembuatan *booklet* dalam bentuk PDF untuk menampilkan *itinerary* perjalanan pelanggan.
4. Pembuatan tab pembayaran dan estimasi biaya.
5. Pembuatan dokumen *invoice* dan *receipt* dalam format PDF.
6. Pengembangan fitur *Setting Perusahaan* untuk mengubah logo, detail, dan rekening perusahaan.
7. Penambahan fitur *Add-ons* serta pembuatan halaman pembayaran baru.

### 3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja selama magang adalah sebagai berikut: 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

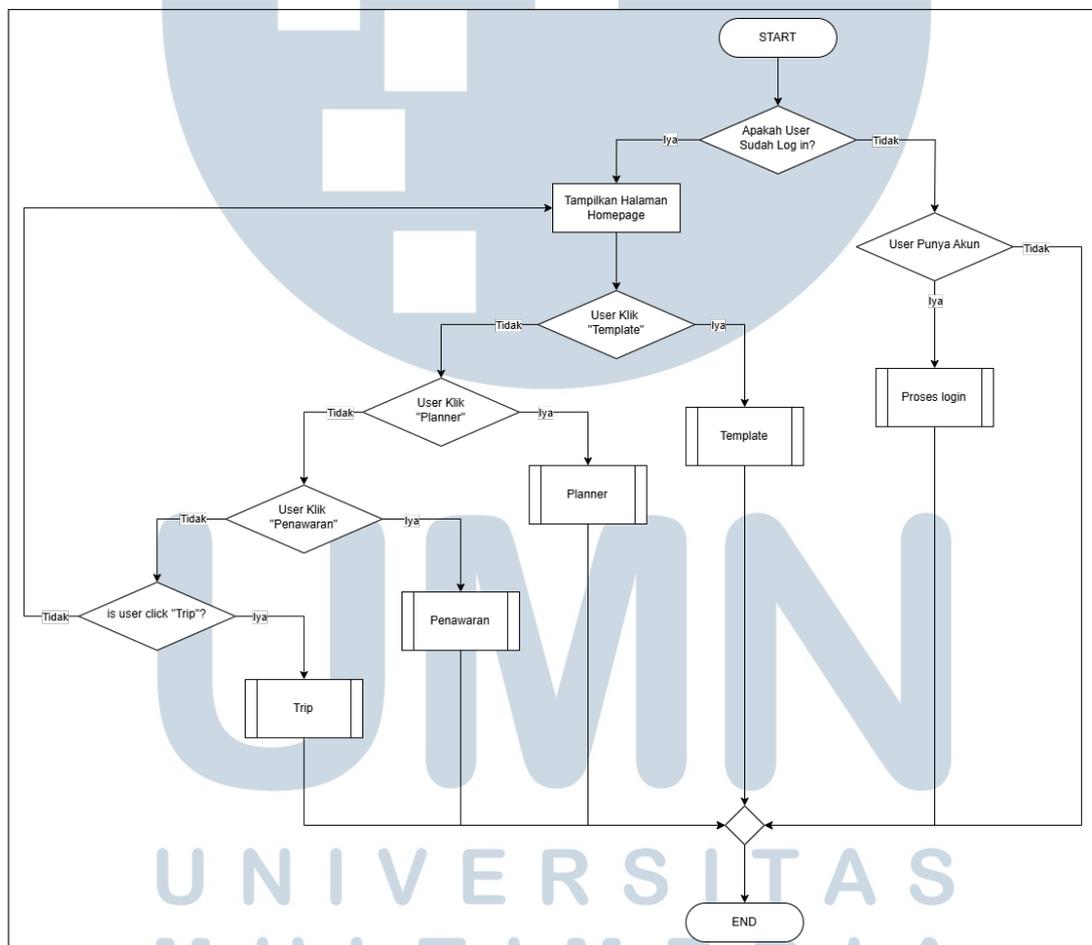
Minggu Ke -	Pekerjaan yang dilakukan
1	Membenarkan <i>bug</i> komponen <i>Free Time</i> , menambahkan serta mengimplementasikan desain yang diberikan pada komponen transportasi dan modal penginapan, menambahkan fitur <i>Upload Photo Gallery</i>
2	Membenarkan <i>bug</i> pada detail komponen item hotel pada ekspor <i>PDF</i> dan menambahkan <i>Expanded Local Guide</i> modal
3	Mengimplementasikan desain yang sudah diberikan untuk pembuatan komponen <i>Receipt</i> dan juga perhitungan serta ekspor ke file <i>PDF</i>
4	Memperbaiki modal tiket dan detail item serta memperbaiki komponen pada <i>booklet</i> , melanjutkan pembuatan <i>Receipt</i> , menambahkan fitur untuk <i>rating</i> bintang hotel, dan memperbaiki piksel yang hilang pada <i>booklet</i>
5	Mempersiapkan dan memahami <i>framework</i> dan <i>flow</i> dari pembuatan aplikasi dan implementasi desain yang sudah diberikan pada komponen <i>Check out Virtual Account</i> di aplikasi <i>mobile</i>
6	Melakukan pembuatan ulang pada tab pembayaran di penawaran serta menambahkan <i>loading skeleton</i> untuk setiap halaman di <i>dashboard</i>
7	Menambahkan dan memperbaiki <i>loading skeleton</i> untuk setiap halaman di <i>dashboard</i>
8	Menambahkan fitur dan komponen untuk pilihan transportasi berupa kapal, memperbaiki <i>Receipt</i> , dan menambahkan <i>skeleton</i> untuk komponen <i>Planner</i>

Minggu Ke -	Pekerjaan yang dilakukan
9	Mengeksplorasi dan memahami cara melakukan <i>unit testing</i> pada <i>framework</i> , melakukan perubahan desain <i>Receipt</i> , membuat dan mengimplementasikan desain yang sudah diberikan pada <i>Invoice</i> untuk diekspor menjadi <i>PDF</i> , serta menambahkan fitur <i>Setting Perusahaan</i>
10	Memperbaiki <i>bug</i> titik temu pada <i>Guide</i> dan menambahkan fitur <i>Add-ons</i> untuk penawaran dan <i>Trip</i>
11	Melanjutkan penambahan fitur <i>Add-ons</i> dan memperbaiki <i>bug</i> data pada <i>Receipt</i>
12	Menambahkan <i>Add-ons</i> pada komponen <i>Table View</i> , menambahkan <i>loading skeleton</i> pada <i>Blog List</i> , <i>Detail Penarikan Dana</i> , <i>Add-ons List</i> , dan <i>Additional Cost List</i> , membuat ulang pengaturan harga dengan menambahkan harga <i>Add-ons</i> dan diskon, serta mengimplementasikan desain baru yang sudah diberikan
13	Menambahkan data pada <i>Invoice</i> setelah penambahan <i>Add-ons</i> dan fitur diskon
14	Menambahkan <i>Guide Booking List</i> pada <i>tab service</i> untuk mempermudah pengguna melihat <i>Booking List</i> tiap <i>Guide</i> , menambahkan fitur filter pada <i>Guide Booking List</i> , memperbaiki <i>bug</i> pada <i>Add-ons</i> , menambahkan <i>Collection GuideBookingList</i> di <i>Firebase</i> , serta memperbaiki harga yang hilang di tab pembayaran
15	Menambahkan dan memperbaiki komponen dan data pada <i>Add-ons</i> , menambahkan harga <i>Add-ons</i> di harga penawaran <i>PDF</i> , melanjutkan perbaikan <i>bug</i> harga yang hilang di tab pembayaran, dan mengeksplor <i>Firebase</i> karena ada <i>issue</i> data hilang
16	Memperbaiki <i>Add-ons</i> , menambahkan fitur <i>sort</i> dan <i>filter</i> pada tabel, memperbaiki <i>footer</i> pada tabel, menambahkan fitur <i>Table View</i> pada komponen <i>Template</i> dan <i>Planner</i> , serta memperbaiki data yang tidak dapat tersimpan

### 3.4 Rancangan Alur User Dashboard Membuat Trip

#### 3.4.1 Home

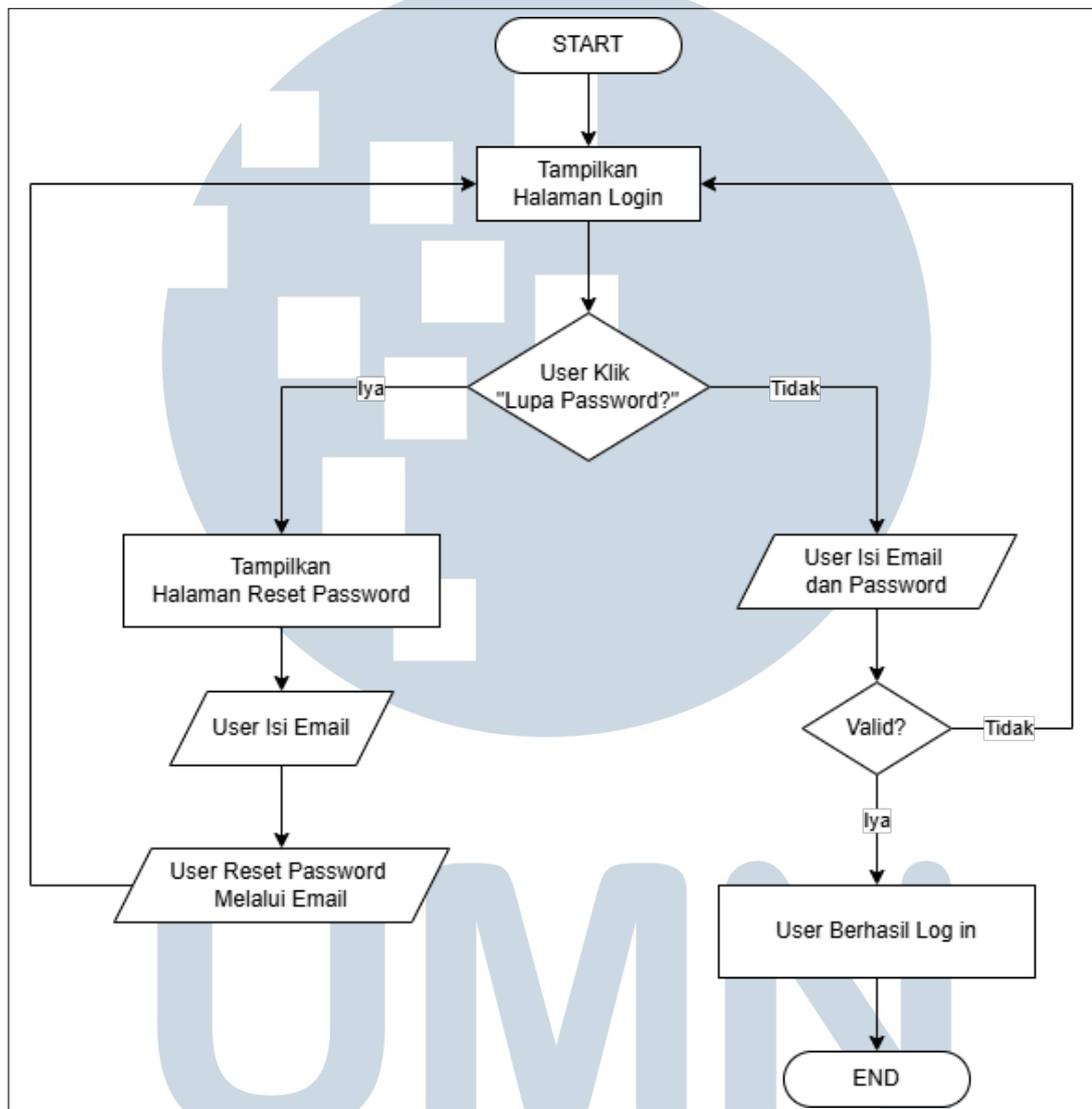
Gambar flowchart 3.1, sebelum dapat mengakses *dashboard*, pengguna harus melalui proses autentikasi terlebih dahulu. Pengguna diminta untuk mengisi formulir berupa email dan kata sandi yang telah terdaftar. Jika belum memiliki akun, pengguna harus melakukan pendaftaran terlebih dahulu. Namun, proses pendaftaran hanya dapat dilakukan melalui situs web lain atau melalui bantuan *admin*.



Gambar 3.1. Flowchart utama

Pada gambar flowchart 3.2, terlihat bahwa pengguna dapat melakukan proses *login* saat pertama kali membuka situs web. Jika pengguna lupa kata sandi, tersedia opsi "*Lupa Password?*" yang dapat digunakan untuk melakukan *reset*. Proses *reset* dilakukan melalui email; pengguna akan menerima tautan

husus untuk mengganti kata sandi. Setelah proses tersebut selesai, pengguna dapat menggunakan kata sandi baru untuk masuk ke dalam *dashboard*.



Gambar 3.2. Flowchart pengguna untuk *login*

Halaman *dashboard* memiliki navigasi di bagian kanan yang memberikan akses ke beberapa fitur untuk pembuatan perjalanan. Sebelum membuat perjalanan, terdapat fitur *Template* yang digunakan untuk menyusun kerangka perjalanan yang nantinya dapat digunakan kembali. Selanjutnya, tersedia fitur *Planner* yang digunakan untuk merancang rencana perjalanan, baik untuk dipasarkan maupun berdasarkan permintaan dari pelanggan.

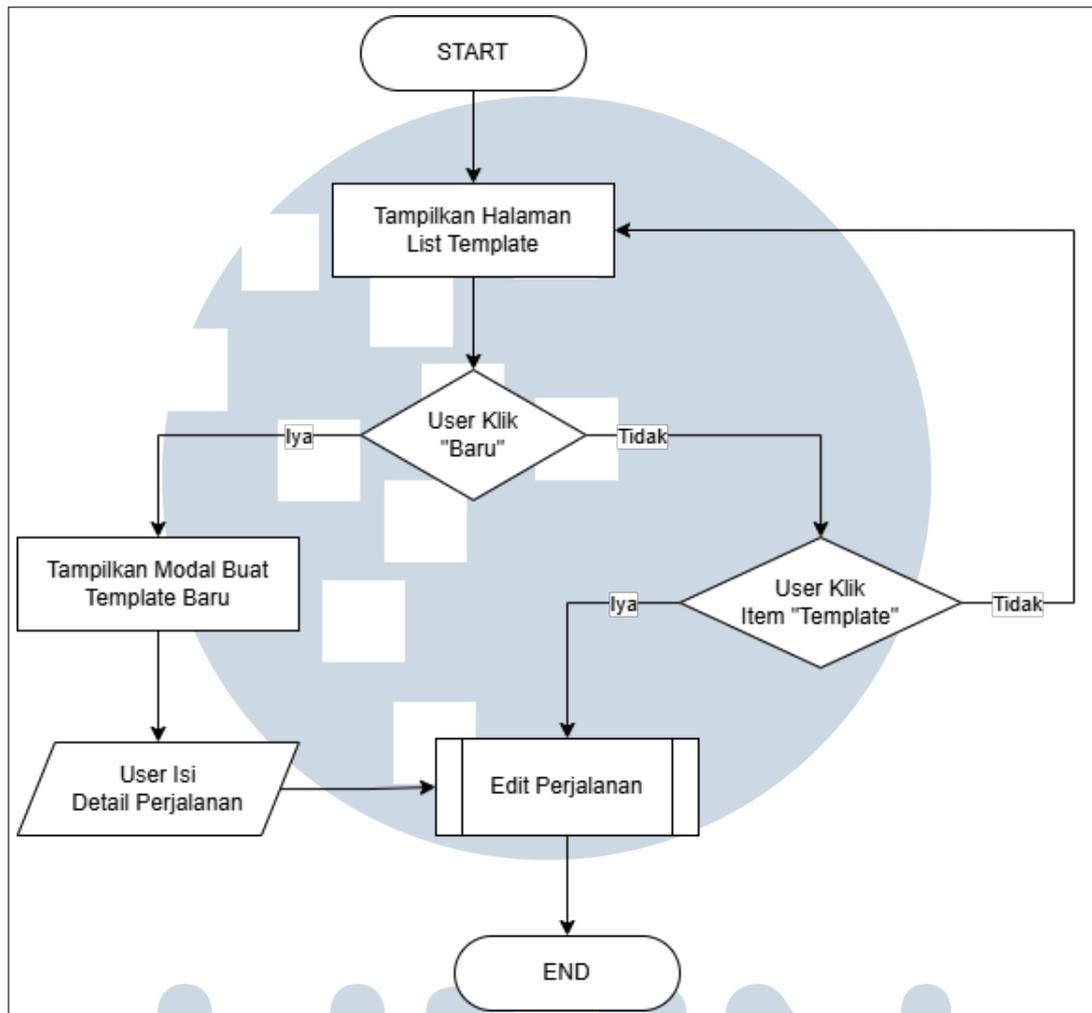
Terdapat pula fitur Penawaran yang berfungsi ketika rencana perjalanan telah sesuai dengan preferensi pelanggan atau ketika terdapat pelanggan yang

tertarik. Fitur ini digunakan untuk menyusun dan menghitung estimasi total biaya perjalanan serta menyusun penawaran harga yang akan diberikan perusahaan kepada pelanggan. Apabila pelanggan menyetujui penawaran tersebut, maka rencana akan menjadi sebuah perjalanan atau *trip*. Tahap akhir mencakup penyelesaian seperti pembelian tiket transportasi, konfirmasi penginapan, dan keperluan lainnya.

### 3.4.2 Template

Alur pembuatan *template* ditunjukkan pada flowchart 3.3. Ketika pengguna masuk ke halaman *template*, yang pertama kali ditampilkan adalah daftar *template* yang telah dibuat sebelumnya. Jika ingin membuat *template* baru, pengguna dapat menekan tombol "Baru" dan mengisi formulir seperti nama *template*, durasi perjalanan, dan deskripsi perjalanan, lalu menekan tombol Simpan. Jika ingin mengedit *template* yang sudah ada, pengguna cukup menekan salah satu item yang terdapat pada daftar.





Gambar 3.3. Flowchart Template untuk membuat *template*

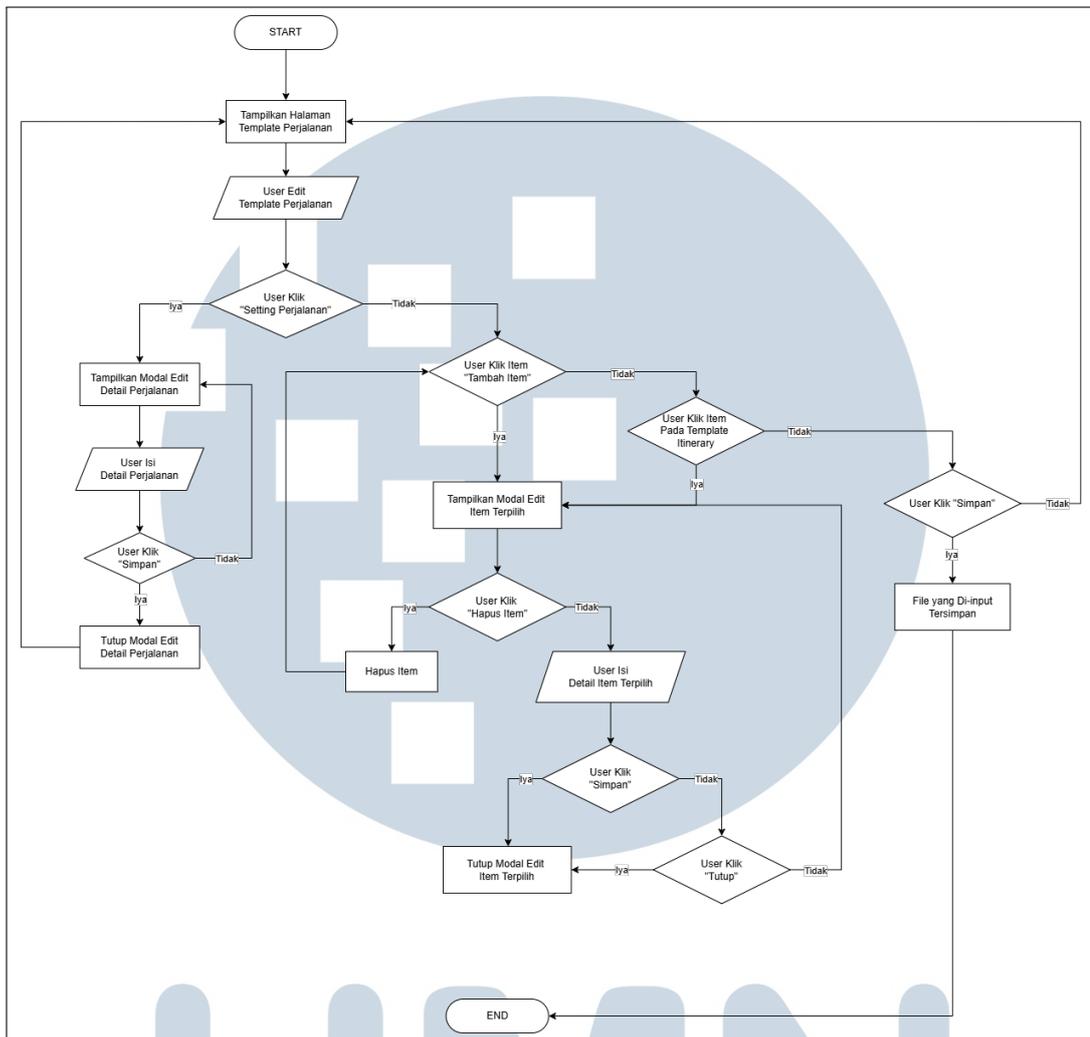
Saat masuk ke detail *template*, pengguna dapat menambahkan item baru dengan menekan "Item Baru" atau memilih salah satu item yang tersedia. Pilihan item yang dapat ditambahkan meliputi:

- Pesawat
- Penginapan
- Taxi Transfer
- Private Transport
- Bus
- Kereta

- Kapal
- Local Guide
- Destinasi
- Free Time
- Event
- Note

Pada flowchart 3.4, ketika pengguna memilih salah satu item, akan muncul jendela *modal* yang dapat diatur sesuai dengan jenis item yang dipilih. Setelah selesai mengisi, pengguna dapat menekan tombol Simpan untuk menyimpan data yang telah diisi. Jika seluruh perjalanan sudah selesai dirancang, pengguna dapat menyimpan seluruh data dan kembali ke halaman utama.

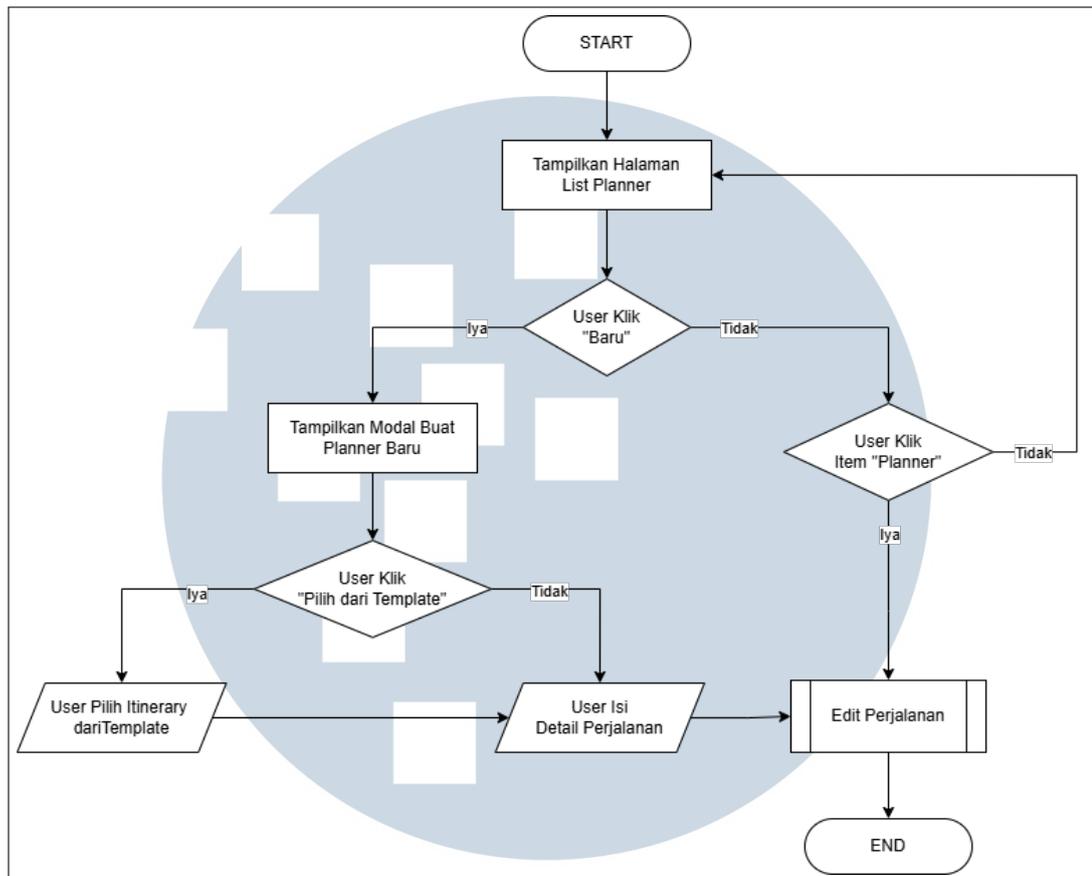
UMMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.4. Flowchart Edit Perjalanan untuk membuat atau mengedit item *template*

### 3.4.3 Planner

*Planner* merupakan fitur yang memiliki fungsi serupa dengan *template*, namun dirancang untuk keperluan yang lebih lanjut dan spesifik. Jika *template* digunakan untuk menyusun kerangka dasar perjalanan, maka *planner* digunakan untuk merancang dan menyesuaikan rencana perjalanan sesuai kebutuhan *customer* atau keperluan promosi. Alur penggunaannya ditunjukkan pada flowchart 3.5. Pengguna dapat membuat *planner* baru dari awal atau berdasarkan *template* yang sudah ada, lalu mengatur detail perjalanan.

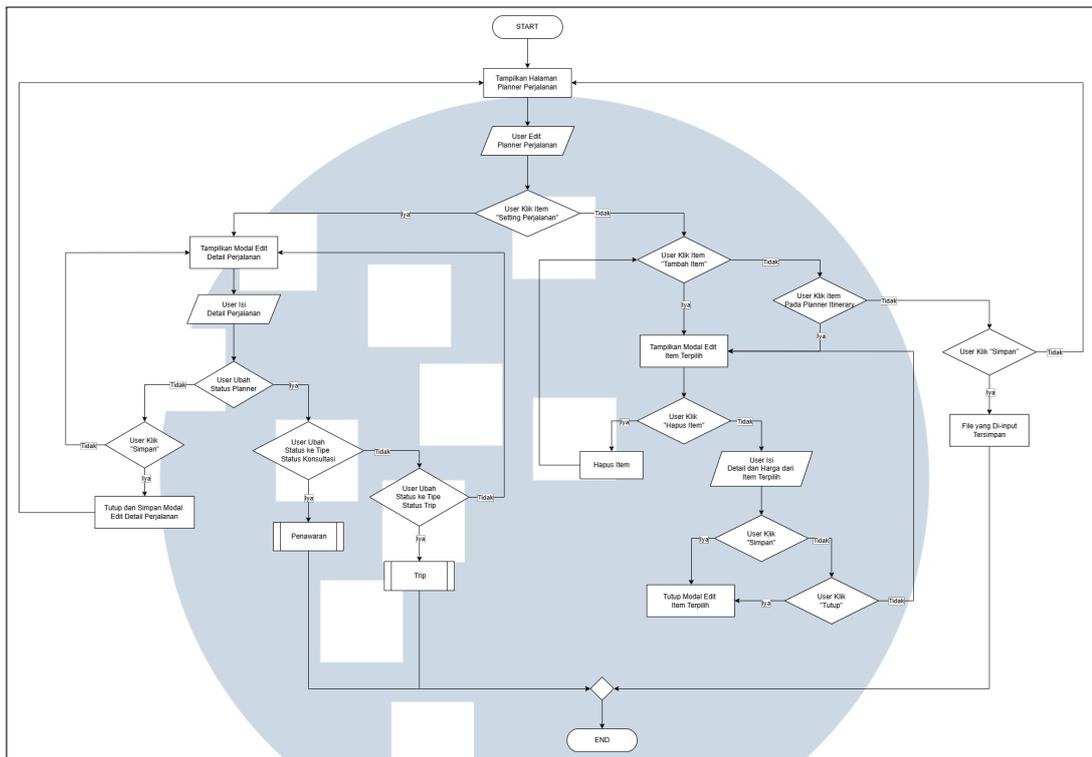


Gambar 3.5. Flowchart Planner untuk membuat *planner*

Pada flowchart 3.6, terlihat bahwa struktur item dalam *planner* hampir sama dengan *template*. Namun, *planner* memiliki pengaturan tambahan berupa status perjalanan yang dapat disesuaikan sepanjang proses perencanaan dan pelaksanaan. Terdapat dua jenis status yang digunakan:

- *offeringStatus*: consultation, qc\_itinerary, offering\_sent, deal, lost, booking
- *tripStatus*: itinerary\_generated, arrival, departure, feedback

Status *offeringStatus* menunjukkan tahapan dalam proses penawaran, mulai dari konsultasi hingga pemesanan atau kegagalan penawaran. Sedangkan *tripStatus* merepresentasikan tahapan perjalanan yang sedang berlangsung atau telah selesai. Pengguna dapat mengubah status tersebut sesuai perkembangan perjalanan atau komunikasi dengan pelanggan.

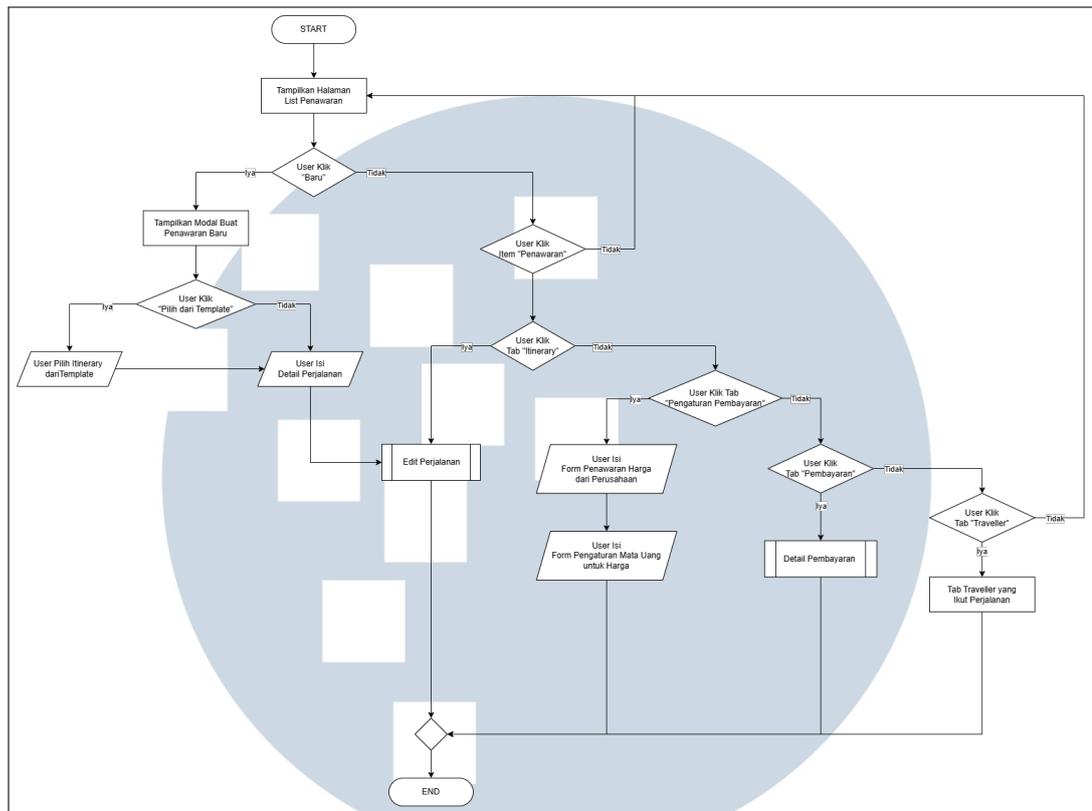


Gambar 3.6. Flowchart Edit Perjalanan untuk membuat atau mengedit item *planner*

### 3.4.4 Penawaran

Penawaran merupakan tahap lanjutan dari proses planner, di mana perencanaan perjalanan mulai disiapkan untuk dikirimkan kepada pelanggan. Struktur item pada penawaran serupa dengan *planner*, namun dengan tambahan pengaturan status dan fitur yang berkaitan dengan transaksi.

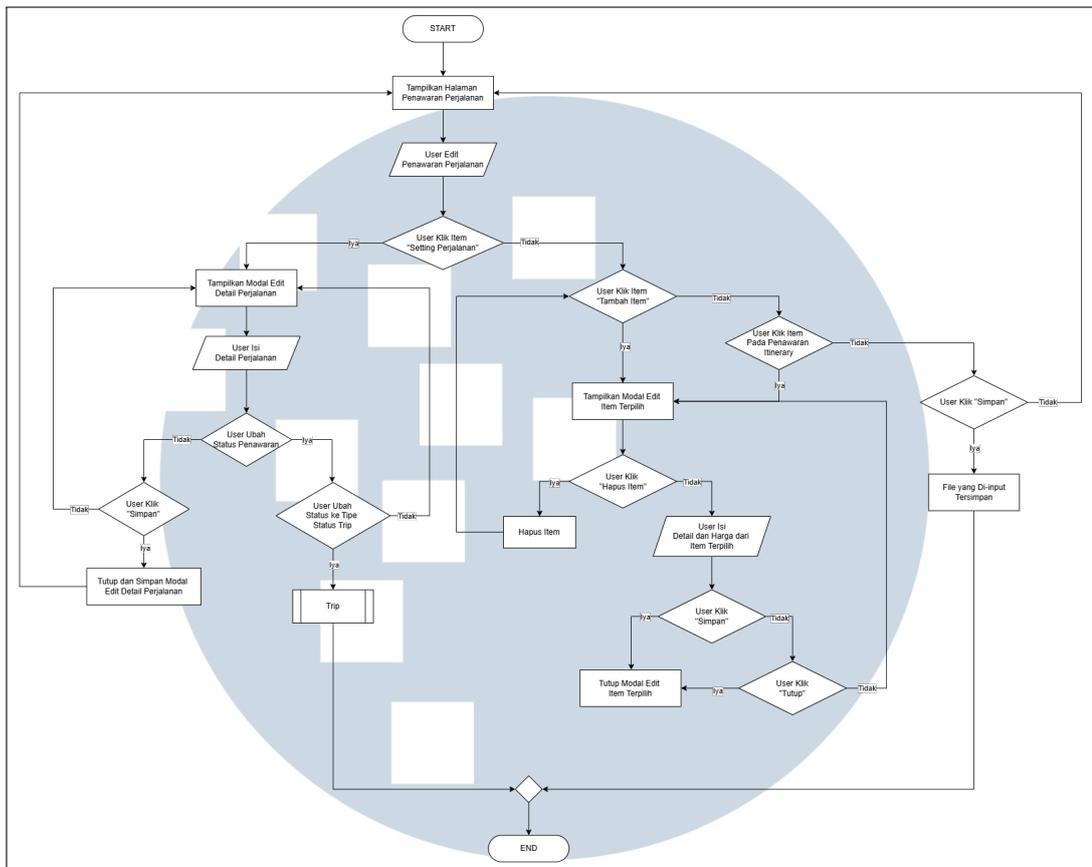
Pengguna dapat mengganti status perjalanan menggunakan `tripStatus` seperti yang digunakan pada *planner*, yaitu: `itinerary.generated`, `arrival`, `departure`, dan `feedback`. Hal ini memungkinkan pelacakan status perjalanan secara lebih jelas setelah tahap penawaran dikirimkan.



Gambar 3.7. Flowchart Penawaran untuk membuat atau mengedit item penawaran

Selain pengaturan itinerary, halaman penawaran juga memiliki beberapa tab tambahan untuk mendukung proses transaksi, yaitu:

- Tab Pengaturan Pembayaran digunakan untuk mengatur mata uang yang digunakan dalam penawaran dan kurs konversi ke Rupiah serta mengatur harga penawaran dari perusahaan untuk pelanggan.
- Tab Pembayaran berisi rincian pembayaran seperti total biaya perjalanan, tenggat waktu pembayaran uang muka (DP), riwayat pembayaran, serta opsi untuk mengunduh *invoice* dan *receipt*.
- Tab Traveller menampilkan daftar peserta perjalanan, yang dapat diisi sesuai dengan data pelanggan atau peserta yang terlibat.
- Tab Itinerary merupakan tampilan utama yang menampilkan susunan rencana perjalanan beserta semua item yang telah ditambahkan.

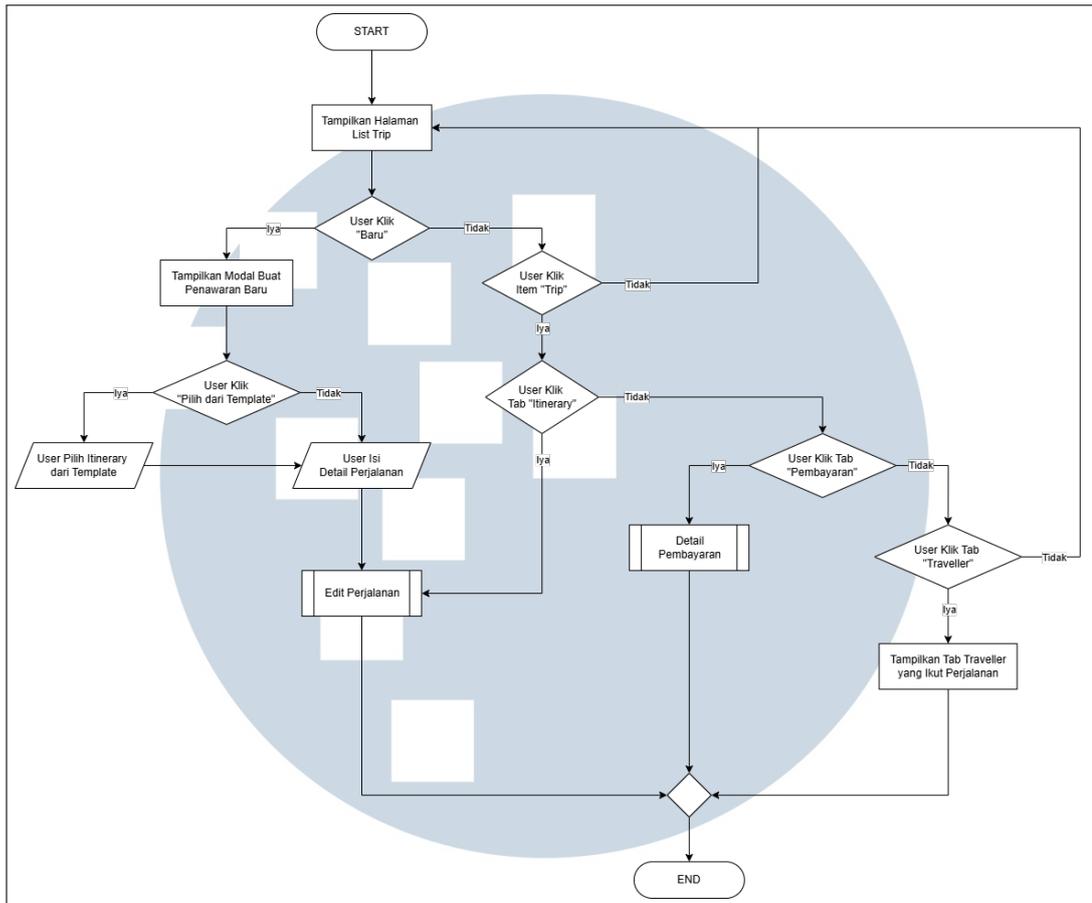


Gambar 3.8. Flowchart Edit Perjalanan untuk membuat atau mengedit item *trip*

### 3.4.5 Trip

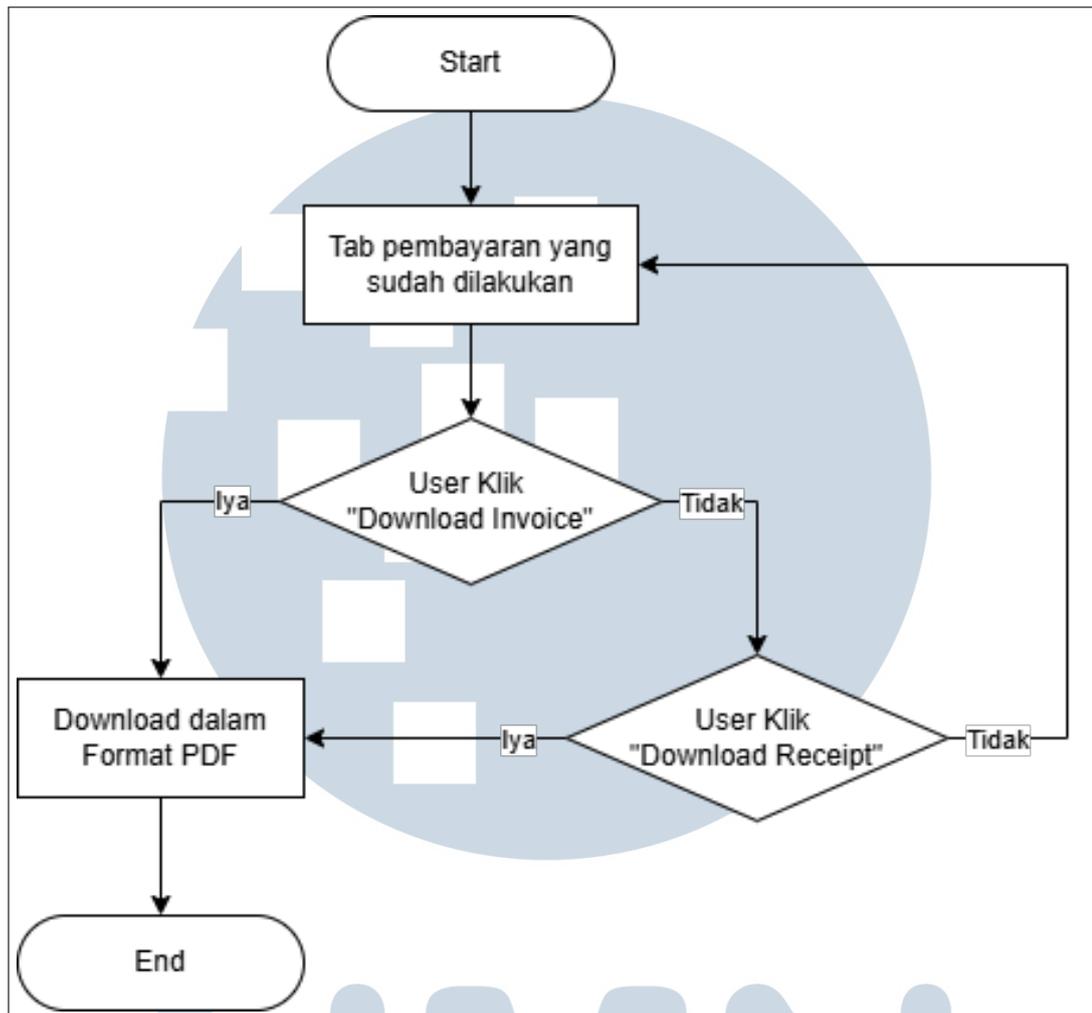
*Trip* merupakan tahap terakhir dari keseluruhan proses perencanaan perjalanan, di mana itinerary yang telah dikonfirmasi dijalankan oleh peserta perjalanan. Secara struktur dan tampilan, halaman ini serupa dengan tahap penawaran seperti pada flowchart 3.9, namun hanya menyisakan fitur-fitur yang relevan untuk pelaksanaan perjalanan.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.9. Flowchart Trip untuk membuat *trip*





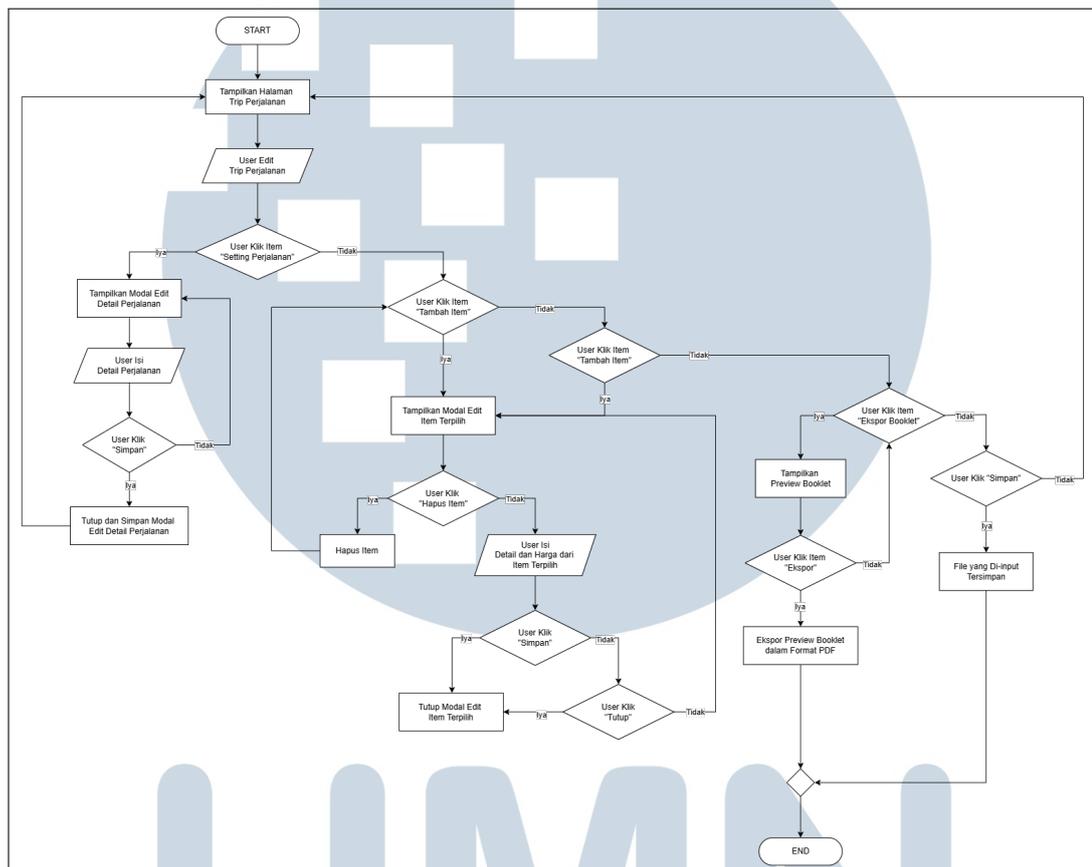
Gambar 3.10. Flowchart Detail Pembayaran untuk pengguna

Tab-tab yang tersedia pada halaman *trip* adalah sebagai berikut:

- Tab Itinerary berfungsi sebagai tampilan utama, menampilkan susunan perjalanan berdasarkan itinerary yang telah dirancang dan disesuaikan sebelumnya.
- Tab Pembayaran menampilkan rincian pembayaran yang telah dilakukan atau masih harus diselesaikan, termasuk histori pembayaran dan dokumen terkait seperti *invoice* atau *receipt*.
- Tab Traveller memuat data peserta perjalanan yang ikut serta dalam pelaksanaan trip.

Dengan tampilan yang lebih ringkas, tahap *trip* berfungsi sebagai pusat kendali perjalanan aktual, memastikan informasi itinerary, pembayaran, dan peserta

dapat diakses dengan mudah selama kegiatan berlangsung, tiket dan kebutuhan lain untuk trip dapat diakses melalui *booklet* yang dapat diekpor dalam bentuk pdf dengan menekan ekspor *booklet* yang berisikan item-item yang terpilih yang ada pada perjalanan untuk menunjukkan itinerary perjalanan pelanggan.



Gambar 3.11. Flowchart Edit Perjalanan untuk membuat atau mengedit item *trip*

### 3.4.6 Implementasi Fitur

#### A Upload Gallery pada Fitur Tambah Item (Penginapan)

Gambar 3.12 memperlihatkan potongan kode dari komponen *Upload Gallery* yang digunakan untuk mengunggah gambar pada beberapa bagian aplikasi, seperti pengaturan kota dan hotel. Fitur ini mendukung dua metode unggah gambar, yaitu dengan melakukan *drag and drop* langsung ke area unggah dengan menggunakan `useDropzone()`, atau memilih gambar secara manual melalui dialog pemilihan file. File-file ini diatasi menggunakan `useState()` dari *props* `images` dan `setImages` dari file yang membutuhkan *photo gallery*.

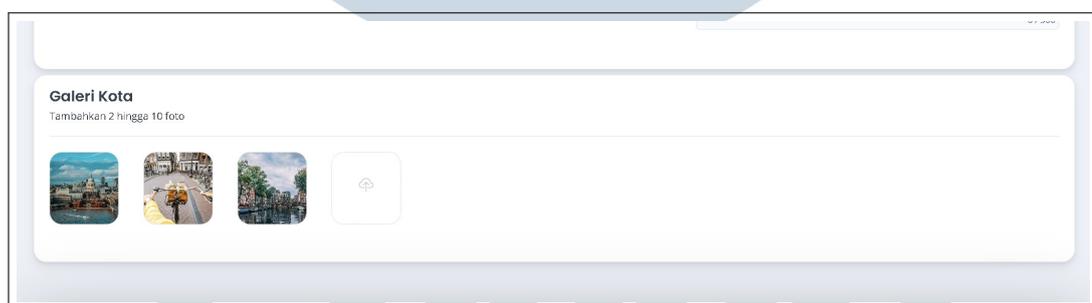
```

82 export function PhotoGalleryContent(props) {
83   const {
84     isLoading,
85     images,
86     setData,
87     photoIndex,
88     setPhotoIndex,
89     setisOpen,
90     props
91   } = props;
92   const inputImage = useRef(null);
93
94   // add image
95   const {
96     getRootProps: getRootPropsAddImage,
97     getInputProps: getInputPropsAddImage
98   } = useDropzone({
99     multiple: true,
100    accept: 'image/*',
101    onDrop: acceptedFiles => {
102      setImages([...(images || []), ...acceptedFiles]);
103    }
104    Promise.all(
105      acceptedFiles.map(file => {
106        return uploadFile('getUploadImage', file);
107      })
108    ).then(valArray => setImages([...(images || []), ...valArray]));
109  });
110
111  // edit image
112  const {
113    // getRootProps: getRootPropsEditImage,
114    // getInputProps: getInputPropsEditImage
115  } = useDropzone({
116    multiple: false,
117    accept: 'image/*',
118    onDrop: acceptedFiles => {
119      const temping = [...images];
120      temping.splice(photoIndex, 1, ...acceptedFiles);
121      setImages(temping);
122    }
123  });

```

Gambar 3.12. Potongan tampilan kode pada editor untuk fitur *Upload Gallery*. Gambar ini menampilkan sebagian kecil dari fungsi pengunggahan gambar ke Firebase Storage.

Gambar 3.13 menunjukkan tampilan antarmuka dari fitur *Upload Gallery* setelah gambar berhasil diunggah. Gambar yang diunggah akan langsung ditampilkan dan dapat dikelola sesuai kebutuhan.



Gambar 3.13. Tampilan fitur *Upload Gallery*

## B Fix PDF Hotel Information Pada Ekspor Booklet Trip

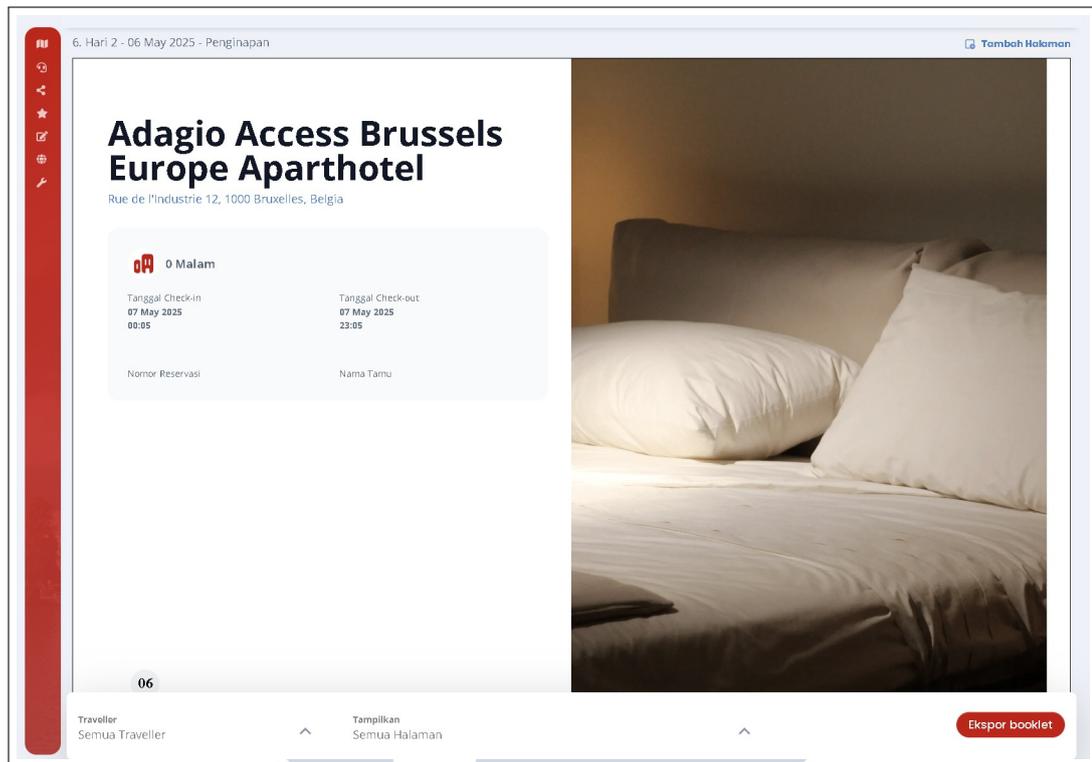
Gambar 3.14 menunjukkan potongan kode yang digunakan untuk menampilkan informasi hotel dalam format PDF. Permasalahan yang dihadapi adalah nama hotel yang terlalu panjang menyebabkan *text overflow*, sehingga mengganggu tampilan. Fungsi `truncatedMiddle()` dibuat karena hal ini, fungsi ini digunakan untuk memotong teks secara otomatis apabila melebihi batas panjang tertentu (misalnya 1250px), dengan menambahkan tanda "..." di bagian tengah teks agar tetap terbaca namun tidak memakan terlalu banyak ruang.

```
65
66
67   style={{
68     display: 'flex',
69     width: scaleValue(713, currentResolution),
70     padding: scaleValue(50, currentResolution),
71     paddingRight: scaleValue(81, currentResolution),
72     padding: scaleValue(100, currentResolution)
73   }}
74
75   <StarRating
76     data={data.lodgingLocation?.star || data.lodgingLocation?.rating}
77   />
78   <GrayStar={false}
79     isPDFView={true}
80   />
81
82   <View
83     style={{
84       margin: scaleValue(16, currentResolution),
85       margin: scaleValue(8, currentResolution)
86     }}
87   />
88   <Text
89     style={{
90       fontSize: scaleValue(68, currentResolution),
91       fontFamily: 'P150Akartasana',
92       fontWeight: 700,
93       color: '#111027'
94     }}
95     {formData?.title &&
96       truncatedMiddle(
97         getShortString(formData.title),
98         250,
99         60,
100         isPDFView
101       )
102   />
103   <Text
104     style={{
105     styles.text_24_400,
```

Gambar 3.14. Potongan kode perbaikan *text overflow* pada item hotel

Gambar 3.15 menunjukkan hasil tampilan ekspor item hotel dalam PDF setelah perbaikan dilakukan. Meskipun efek pemangkasan teks (“...”) tidak tampak secara eksplisit pada gambar ini, fungsi tersebut tetap aktif dan akan bekerja jika nama hotel melebihi panjang yang ditentukan.





Gambar 3.15. Tampilan hasil ekspor item hotel

### C Ekspor Receipt PDF Pada Tab Pembayaran

Gambar 3.16 menunjukkan potongan kode untuk pembuatan *receipt* dalam format PDF. Proses pembuatan PDF ini menggunakan library `@react-pdf/renderer`, yang memungkinkan pembuatan dan pengaturan komponen PDF secara deklaratif dengan sintaks mirip React. Dalam implementasinya, komponen PDF menerima props berupa data transaksi yang telah dilakukan oleh pengguna. Data ini mencakup informasi detail pembayaran seperti ID pembayaran, tanggal, metode, dan total pembayaran, yang kemudian ditampilkan dalam layout terstruktur di dalam PDF.

```
19 export const ReceiptPDF = ({
20   formData,
21   records,
22   company,
23   recordId,
24   brand
25 }) => {
26   return (
27     <View
28       style={{
29         width: '100%',
30         height: '100%'
31       }}
32     >
33       <View
34         style={{
35           display: 'flex',
36           flexDirection: 'column',
37           padding: '10px',
38           gap: 10
39         }}
40       >
41         <Text>
42           {`Header`}
43         </Text>
44         <Image
45           src={logo}
46           style={{ width: '100px', height: '100px', objectFit: 'cover' }}
47         </Image>
48         <View
49           style={{
50             margin: '10px',
51             display: 'flex',
52             flexDirection: 'column'
53           }}
54         >
55           <Text
56             style={{
57               fontWeight: 'bold',
58               fontSize: '16px'
59             }}
60           >
61             {formData?.companyDetail?.name}
62             <br/>
63             {formData?.companyDetail?.name}
64           </Text>
65           <Text
66             style={{
67               margin: '10px',
68               fontSize: '12px'
69             }}
70           >
71             {`Footer`}
72         </View>
73       </View>
74     </View>
75   )
76 }
```

Gambar 3.16. Potongan kode untuk ekspor receipt

Gambar 3.17 memperlihatkan tampilan akhir hasil ekspor PDF receipt. Semua informasi pembayaran ditampilkan dengan rapi dan sesuai format yang telah ditentukan, sehingga memudahkan pengguna untuk mencetak atau menyimpan bukti transaksi secara digital.





PT Gue Eksplorasi Terust  
BSD Green Office Park 1, South Tower 6th Floor  
Tangerang, 15345

☎ (+62)811909740 ✉ testadmin@getgoing.co.id  
🌐 www.getgoing.co.id3

PEMBAYARAN DITERIMA

Test Xendit

Record ID: 22407143 / Plan ID: d0f6e6

ID Pembayaran

Tanggal Pembayaran

20 Apr 2025, 06:13

Metode Pembayaran

VISA Credit Card

Jumlah Pembayaran  
(Nominal)

Rp 500.000  
( LIMA RATUS RIBU )



Eksplor dan Temukan  
Hidden Gems di Eropa

Bikin Rencana Perjalanan Lebih Mudah.  
Pilih Destinasi, Transportasi, Akomodasi Sesuka Hati dan  
Buat Liburan dengan Caramu Sendiri.

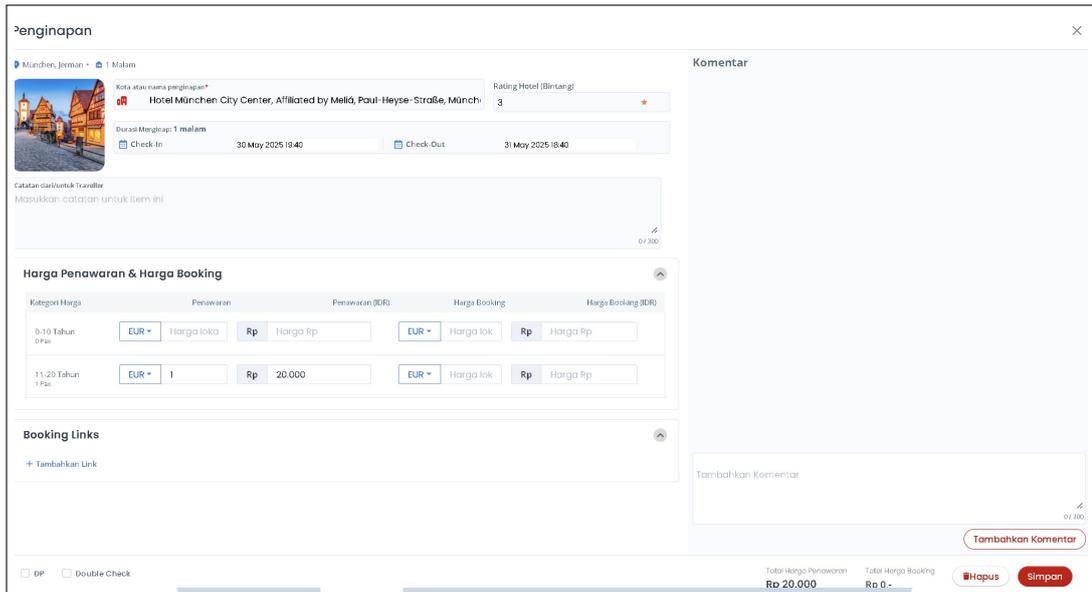
Download  
Aplikasi GetGoing



NUSANTARA

Gambar 3.17. Tampilan hasil ekspor receipt

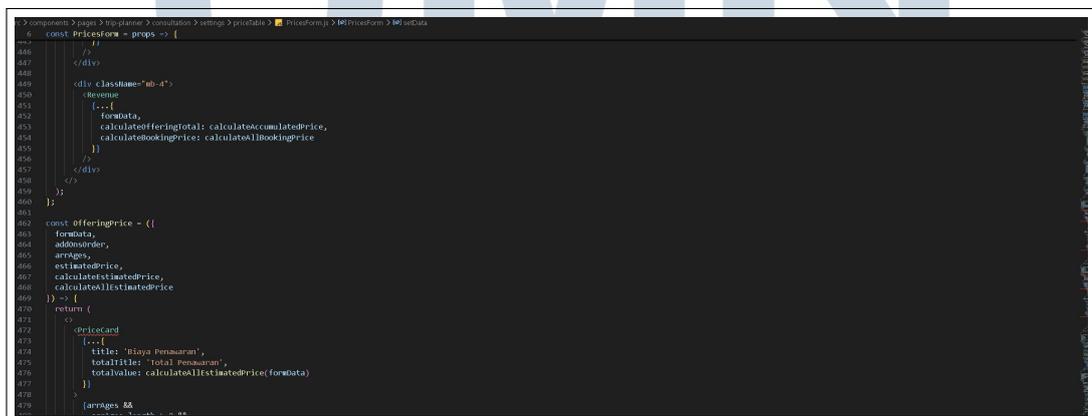




Gambar 3.19. Tampilan form penilaian hotel

## E Tab Pengaturan Pembayaran Pada Fitur Penawaran

Gambar 3.20 menunjukkan potongan kode pembuatan ulang tab Pengaturan Pembayaran. Komponen utama yang ditampilkan adalah `OfferingPrice`, yang digunakan untuk menampilkan dan menghitung biaya penawaran yang diberikan kepada pelanggan. Komponen ini berisi dua bagian utama, yaitu Harga Penawaran (harga asli yang ditawarkan kepada customer) dan Harga `GetGoing` (harga khusus dari pihak `GetGoing`). Sistem secara otomatis menghitung margin atau selisih antara kedua harga tersebut sebagai bahan pertimbangan efisiensi biaya.



Gambar 3.20. Potongan kode untuk tab pengaturan pembayaran

Pada Gambar 3.21, tampilan tab pengaturan pembayaran masih sangat

seederhana. Tidak terdapat detail seperti subtotal, diskon, atau tampilan total yang lebih komprehensif. Informasi hanya terbatas pada margin dan harga per pax, tanpa adanya kalkulasi lanjutan atau opsi tambahan.

Itinerary				Pengaturan Harga				Pembayaran				Traveller			
<b>Biaya Penawaran</b>								<b>Down Payment Minimum</b>							
- tahun	Rp. 400.247	x 2 Pax	Rp. 800.494	- tahun	Rp. 0,-	x 2 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-
- tahun	Rp. 460.125	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-
- tahun	Rp. 3	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-
- tahun	Rp. 4	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-
<b>Total Penawaran</b>				<b>Rp. 800.494</b>				<b>Total DP Minimum</b>				<b>Rp. 0,-</b>			
<b>Harga GetGoing</b>															
Tipe Harga	Margin		Harga (per Pax)		Harga (Total)										
- Tahun 2 Pax	-96.92 %	Rp.	12.345	Rp. 24.690											
- Tahun 0 Pax	0 %	Rp.		Rp. 0,-											
- Tahun 0 Pax	0 %	Rp.		Rp. 0,-											
- Tahun 0 Pax	0 %	Rp.		Rp. 0,-											
<b>Total</b>	<b>Total Margin</b>	<b>-96.92 %</b>	<b>Total Harga</b>		<b>Rp. 24.690</b>										
<b>Biaya Booking</b>															
- tahun	Rp. 0,-	x 2 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-
- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-
- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-
- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	- tahun	Rp. 0,-	x 0 Pax	Rp. 0,-
<b>Total Booking</b>				<b>Rp. 0,-</b>				<b>Total Booking</b>				<b>Rp. 0,-</b>			

Gambar 3.21. Tampilan tab pengaturan pembayaran (sebelum perbaikan)

Setelah dilakukan perbaikan seperti terlihat pada Gambar 3.22, tampilan menjadi jauh lebih lengkap dan informatif. Telah ditambahkan elemen-elemen penting seperti:

1. Subtotal, yang menjumlahkan total harga sebelum diskon
2. Fitur diskon, baik berdasarkan nominal maupun jenis diskon
3. Perhitungan total margin, total harga, dan DP (down payment)
4. Rincian biaya booking yang lebih jelas per kategori usia

Perubahan ini mempermudah pihak trip manager dalam melakukan pengelolaan dan pengecekan harga akhir.

### Harga

Biaya Penawaran				Down Payment Minimum			
0-10 Tahun	Rp. 0,-	x 0 Pax	Rp. 0,-	0-10 Tahun	Rp. 0,-	x 0 Pax	Rp. 0,-
11-20 Tahun	Rp. 20.000	x 1 Pax	Rp. 20.000	11-20 Tahun	Rp. 0,-	x 1 Pax	Rp. 0,-
<b>Total Penawaran</b>			<b>Rp. 20.000</b>	<b>Total DP Minimum</b>			<b>Rp. 0,-</b>

#### Harga GetGoing

Harga yang ditawarkan GetGoing

Kategori Harga	Margin	Harga (per Satuan)	Harga (Total)	DP (per Satuan)	DP (total)
0-10 Tahun 0 pax	0.00 %	<input type="text"/>	Rp 0	<input type="text"/>	Rp 0
11-20 Tahun 1 pax	0.00 %	<input type="text"/>	Rp 0	<input type="text"/>	Rp 0
<b>Subtotal</b>	-100.00 %		Rp 0,-		Rp 0,-
Discount	Discount Type: <input type="text"/>	<input type="text"/>	Rp 0,-		
<b>Total</b>	Total Margin <b>-100.00%</b>		Total Harga <b>Rp 0,-</b>		Total DP <b>Rp 0,-</b>

Biaya Booking			
0-10 Tahun	Rp. 0,-	x 0 Pax	Rp. 0,-
11-20 Tahun	Rp. 0,-	x 1 Pax	Rp. 0,-
<b>Total biaya booking (beban perjalanan)</b>			<b>Rp. 0,-</b>

Net Revenue (%) Net Revenue (Rp)

Batalkan
Simpan Perubahan

Gambar 3.22. Tampilan tab pengaturan pembayaran (setelah perbaikan)

## F Ekspor Invoice PDF pada Tab Pembayaran

Pada Gambar 3.23, ditampilkan potongan kode implementasi fitur ekspor *invoice* dalam format PDF menggunakan library `@react-pdf/renderer`. Komponen PDF dikembangkan dengan pendekatan berbasis React, memanfaatkan struktur *component-based* seperti `<Document>`, `<Page>`, `<Text>`, dan `<View>`, yang disediakan oleh *library* tersebut.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

```

47 export const InvoicePDF = ({
92   const invoice = [
98     ],
99   },
100   { display: 'jumlah',
101     width: columnWidth.col3,
102     textStyle: textStyle.col3
103   },
104   { display: 'jumlah', width: columnWidth.col4, textStyle: textStyle.col4 },
105   },
106   { display: 'harga', width: columnWidth.col5,
107     textStyle: textStyle.col5
108   },
109   { display: 'harga', width: columnWidth.col6, textStyle: textStyle.col6 }
110 ]
111 }
112 const totalPaymentDue = formatCurrency(totalPrice, remainingBill || 0);
113 // ini kalau mau diotin sesuai detail pembayarannya
114 // const companyLogo = formatCurrency(companyDetail.logo) || logo;
115 const companyLogo = logo;
116 return (
117   <View
118     style={{
119       position: 'relative',
120       width: '100%',
121       height: '100%'
122     }}
123   >
124     <View
125       style={{
126         display: 'flex',
127         flexDirection: 'column',
128         padding: 40,
129         paddingTop: 0,
130         paddingBottom: 100,
131         width: '100%'
132       }}
133     >
134       <Header />
135       <View
136         style={{
137           width: '100%',
138           fontSize: size.medium
139         }}
140       >
141         <Image
142           src={replaceImage(companyLogo)}
143           style={{ width: '100px', height: '48.5px', objectFit: 'contain' }}
144         />
145       </View>
146     </View>
147   </View>
148 )

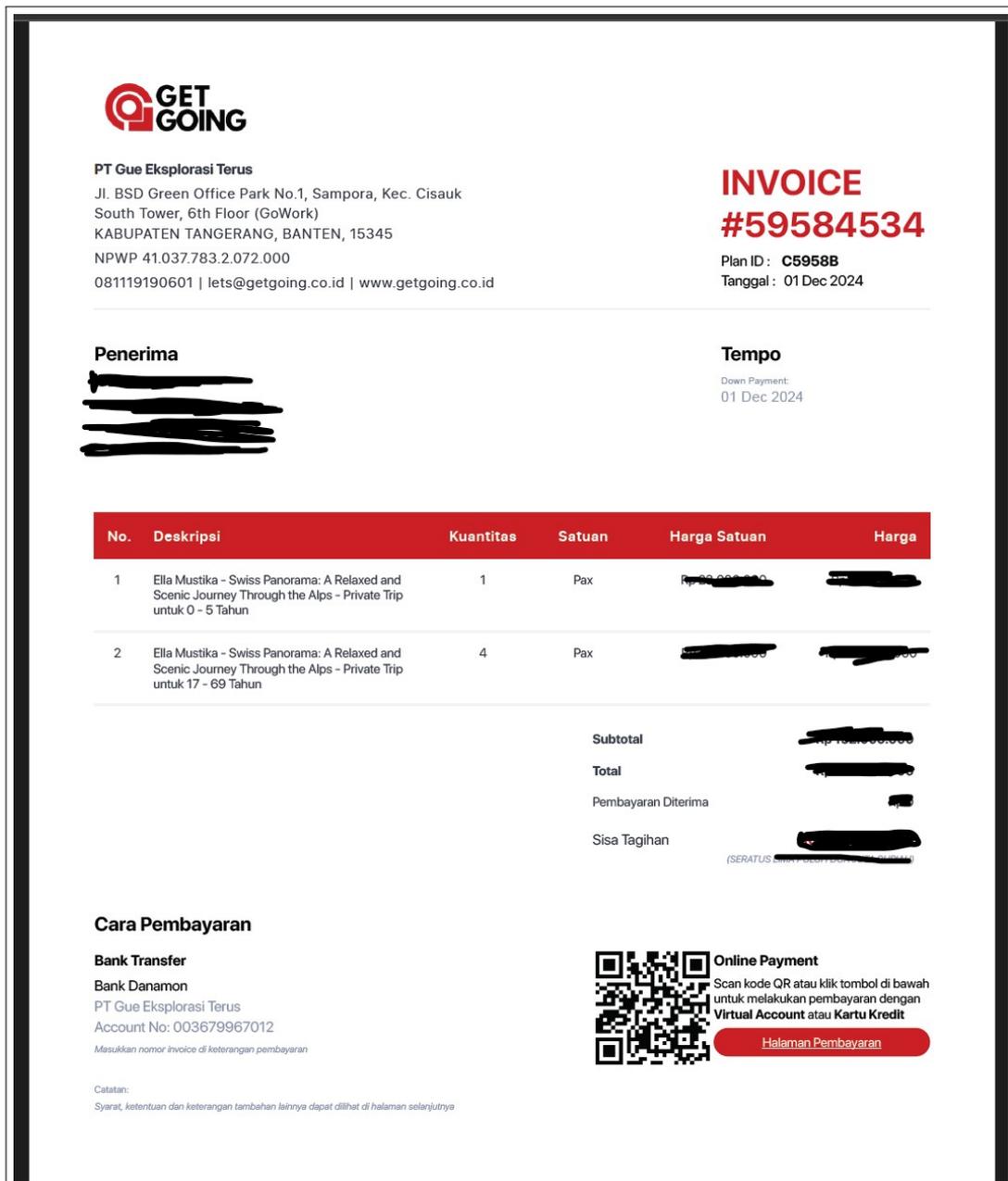
```

Gambar 3.23. Potongan kode untuk ekspor invoice

Data yang digunakan pada *invoice* diambil langsung dari pengaturan harga yang telah diisi sebelumnya oleh *trip manager*. Informasi tersebut mencakup rincian kategori harga, jumlah pax, subtotal, diskon, serta detail pembayaran yang sudah dilakukan (sisa tagihan dan total yang telah dibayarkan). Komponen PDF menerima data tersebut melalui *props*, kemudian dirender secara dinamis ke dalam layout yang telah disesuaikan.

Gambar 3.24 menampilkan hasil akhir dari proses ekspor PDF. Layout dibuat menyerupai dokumen profesional, dengan pemisahan elemen berdasarkan blok harga, jumlah orang, total pembayaran, serta informasi pembayaran yang telah dilakukan. Untuk styling, digunakan pendekatan `StyleSheet.create()` yang disediakan oleh `@react-pdf/renderer`, guna mengatur posisi, margin, dan format teks secara konsisten.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

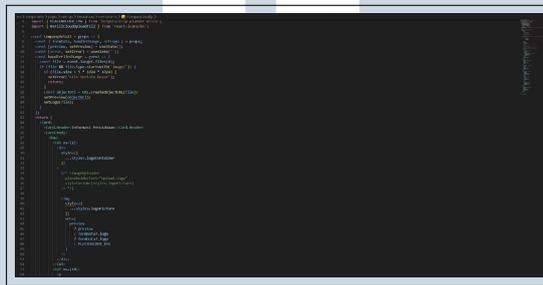


Gambar 3.24. Tampilan hasil ekspor invoice

Dengan penggunaan @react-pdf/renderer, proses pembuatan dan ekspor PDF dapat dilakukan sepenuhnya di sisi klien (*client-side rendering*), tanpa ketergantungan terhadap proses backend tambahan. Hal ini memudahkan pengguna dalam mengakses dan mengunduh dokumen invoice secara langsung dari browser.

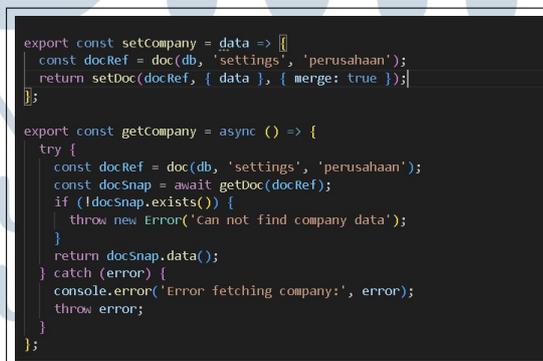
## G Pengaturan Perusahaan

Pada potongan kode di Gambar 3.25, state `useState` digunakan untuk menyimpan data dari form pengaturan perusahaan yang diisi oleh pengguna. Fungsi `handleFileChange` berperan dalam menangani perubahan file logo perusahaan, yang nantinya akan diupload dan disimpan. Data yang telah dimasukkan pada form dapat disimpan ke database Firebase menggunakan fungsi `setCompany()` seperti dijelaskan pada Gambar 3.26.

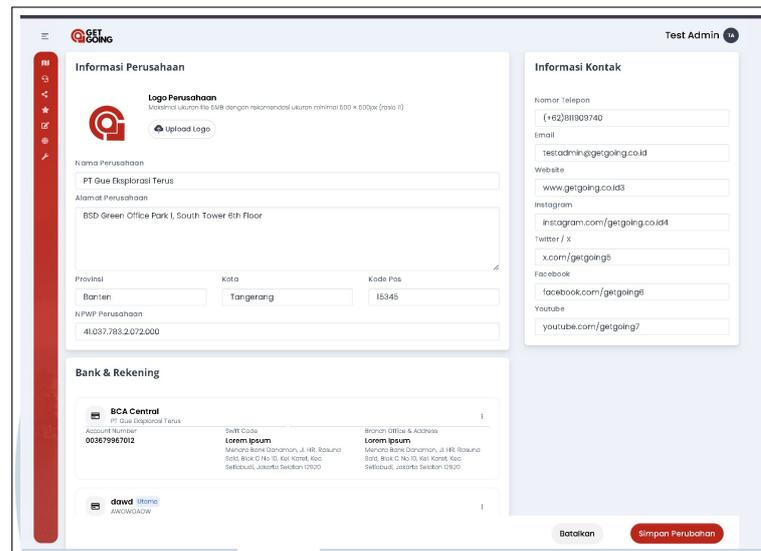


Gambar 3.25. Potongan kode fitur pengaturan perusahaan menggunakan `useState` untuk menyimpan data form serta fungsi `handleFileChange` untuk mengelola perubahan logo perusahaan.

Pada Gambar 3.26, fungsi `setCompany()` digunakan untuk menyimpan data pengaturan perusahaan ke Firestore pada dokumen dengan koleksi `settings` dan dokumen perusahaan. Data yang disimpan berupa JSON yang berisi informasi seperti metode pembayaran, detail lokasi, nama perusahaan, logo, dan lain-lain. Selain itu, fungsi `getCompany()` juga dibuat untuk mengambil data perusahaan tersebut dari Firestore, sehingga data dapat digunakan pada fitur lain seperti invoice dan receipt.



Gambar 3.26. Kode untuk menyimpan data pengaturan perusahaan dalam bentuk JSON ke dalam Firestore menggunakan `doc(db, 'settings', 'perusahaan')`.



Gambar 3.27. Tampilan antarmuka fitur pengaturan perusahaan yang memungkinkan pengguna mengubah detail perusahaan termasuk logo dan informasi terkait lainnya.

## H Item Add-Ons

Fitur *add-ons* yang sebelumnya telah diimplementasikan mengalami pengembangan ulang untuk meningkatkan fungsionalitas dan skalabilitas. Dalam pengembangan terbaru, fitur ini dibagi menjadi dua komponen utama, yaitu *add-ons ditawarkan* dan *add-ons dipesan*.

*Add-ons ditawarkan* berfungsi sebagai *template* item tambahan yang dapat ditawarkan kepada *customer*. Setiap entri dalam *template* ini memiliki atribut harga dan kapasitas maksimal pax, yang menunjukkan batas jumlah pengguna yang dapat memilih *add-on* tersebut dalam satu *trip*.

Ketika salah satu item ditambahkan ke dalam itinerary, data dari *add-ons ditawarkan* akan dipindahkan ke *add-ons dipesan*. Pada bagian ini, *trip manager* dapat menentukan jumlah pax yang akan memesan *add-on* tersebut. Validasi dilakukan untuk memastikan jumlah pax tidak melebihi kapasitas maksimal yang telah ditentukan sebelumnya di *template*.

Pengaturan harga untuk setiap item yang dipilih dilakukan secara dinamis. Penggunaan `useEffect()` dalam implementasi React digunakan untuk mengelola dan memperbarui data harga secara real-time, tidak hanya terbatas pada *add-ons*, tetapi juga mencakup data *offering*, *harga getgoing*, *booking*, dan data dinamis lainnya yang berkaitan dengan proses pemesanan.



```

143 //
144 // const BookingList = () => {
145 //   function handleDoc(doc, params) {
146 //     if (str === 'new') {
147 //       addDoc();
148 //       return;
149 //     }
150 //     const columns = [
151 //       // id
152 //       {
153 //         accessor: 'id',
154 //         header: 'ID',
155 //         headerProps: { className: '' },
156 //         cellProps: {
157 //           className: 'py-2'
158 //         },
159 //         cell: rowData => {
160 //           const { id, tripId } = rowData.row.original;
161 //           return (
162 //             <div>
163 //               <div
164 //                 className="text-secondary p-0"
165 //                 onClick={() => onPlanClick(tripId)}
166 //               >
167 //                 Plan ID {getShortId(tripId) || ''}
168 //               </div>
169 //               <div className="font-2 p-0">
170 //                 Trip ID <span className="font-bold">{getShortId(id)}</span>
171 //               </div>
172 //             </div>
173 //           );
174 //         },
175 //       },
176 //     ];
177 //     // author
178 //     {
179 //       accessor: 'guide',
180 //       header: 'Guide',
181 //       headerProps: { className: 'pe-1' },
182 //       cellProps: {
183 //         className: 'py-2',
184 //         onClick: onPlanClick
185 //       },
186 //       cell: rowData => {
187 //         const {
188 //           authorName,
189 //           authorAvatar,
190 //           authorEmail,
191 //           city = ''

```

Gambar 3.30. Potongan kode untuk fitur *Guide Booking List*

Variabel `columns` kemudian diproses dengan metode `map()` untuk menyesuaikan setiap kolom dengan *props* yang sesuai, sehingga data dapat ditampilkan secara dinamis per kolom.

Data untuk tabel ini diambil dari *database* Firebase. Pengambilan data dilakukan menggunakan fungsi `getAllBookingListData()`, yang mengambil dokumen dari referensi `doc(db, 'aggregateList', 'guideBookingList')` seperti yang ditunjukkan pada Gambar 3.31. Koleksi `guideBookingList` disimpan dalam *collection* `aggregateList`, dan berbentuk objek yang berisi data dalam format JSON.

```

export const getAllBookingListData = async () => {
  const docRef = doc(db, 'aggregateList', 'guideBookingList');

  return getDoc(docRef).then(snap => {
    if (snap.exists()) {
      const data = snap.data();
      return data || {};
    } else {
      return {};
    }
  });
};

```

Gambar 3.31. Pengambilan data booking dari Firebase

ID	Guide	Durasi & Waktu	PIC Traveller	Booking Type	Trip Manager	Status	Total Fee	Aksi
Plan ID dfa188 Item ID 896234	Desi Madrid, Spanyol	Min, 17 Agt 2025 06:40 - 21:40 (15 jam)	-	Dashboard 15/05/2025, 13:21	-	Upcoming	Rp 4.780.000 Rp 305.000 / jam	
Plan ID b089dc Item ID 4445f6	Jahid Milan, Italia	Sen, 08 Jul 2025 16:00 - 20:00 (10 jam)	Mohammad Rizal Situru	Dashboard 27/05/2025, 17:31	-	Upcoming	Rp 3.225.000 Rp 305.000 / jam	
Plan ID b089dc Item ID a0488a	Jahid Milan, Italia	Sen, 07 Jul 2025 12:50 - 17:50 (5 jam)	Mohammad Rizal Situru	Dashboard 27/05/2025, 17:31	-	Upcoming	Rp 1.680.000 Rp 305.000 / jam	
Plan ID c1b211 Item ID 44b52a	Paris, Prancis	Min, 22 Jun 2025 09:00 - 17:00 (8 jam)	rany.wahyu@gmail.com	Dashboard 02/06/2025, 08:25	-	Upcoming	Rp 0.- Rp 0.- / jam	
Plan ID c1b211 Item ID 51451a	Nazar Amsterdam, Belanda	Kam, 19 Jun 2025 09:00 - 17:00 (8 jam)	rany.wahyu@gmail.com	Dashboard 02/06/2025, 08:25	-	Upcoming	Rp 2.645.000 Rp 305.000 / jam	
Plan ID a54769 Item ID 0098ba	Budapest, Hungaria	Kam, 19 Jun 2025 06:00 - 11:00 (5 jam)	-	Dashboard 26/05/2025, 12:37	-	Upcoming	Rp 0.- Rp 0.- / jam	
Plan ID c1b211 Item ID b64910	Nazar Amsterdam, Belanda	Rab, 18 Jun 2025 16:00 - 21:00 (5 jam)	rany.wahyu@gmail.com	Dashboard 02/06/2025, 08:25	-	Upcoming	Rp 1.730.000 Rp 305.000 / jam	
Plan ID a54769 Item ID ca0ab9	Theo wina, acbiba	Sen, 16 Jun 2025 06:30 - 11:30 (5 jam)	-	Dashboard 26/05/2025, 12:37	-	Upcoming	Rp 1.730.000 Rp 305.000 / jam	
Plan ID e5e549 Item ID ba99e3	Hudan Roma, Italia	Jum, 06 Jun 2025 08:30 - 18:30 (10 jam)	Hendra Adi hendra adi.253@gmail.com	Dashboard 19/05/2025, 15:36	-	Upcoming	Rp 3.400.000 Rp 320.000 / jam	
Plan ID e5e549 Item ID 5544d3	Hudan Roma, Italia	Kam, 05 Jun 2025 12:00 - 19:00 (7 jam)	Hendra Adi hendra adi.253@gmail.com	Dashboard 19/05/2025, 15:36	-	Upcoming	Rp 2.440.000 Rp 320.000 / jam	

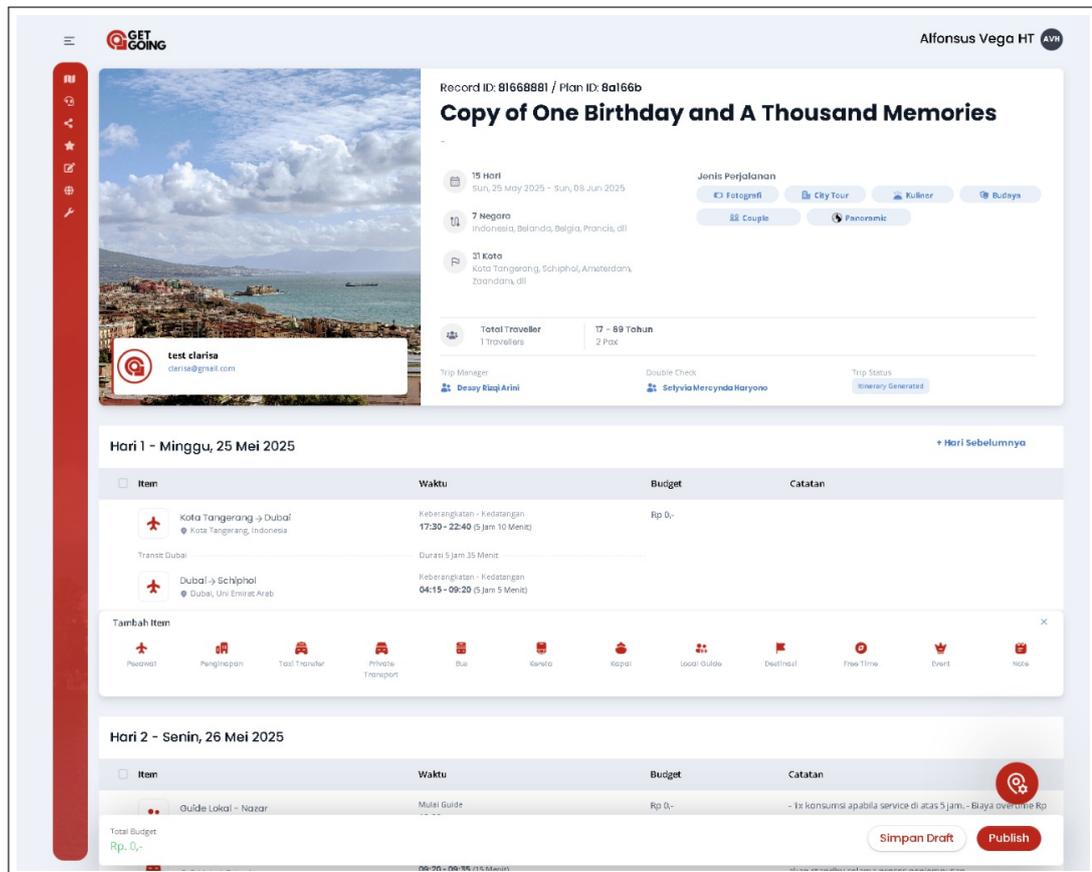
Gambar 3.32. Tampilan fitur *Guide Booking List*

## J Table View Planner dan Template

Fitur *Table View* telah dikembangkan sebagai sarana untuk mempermudah *trip manager* dalam menyusun rencana *itinerary*. Namun, pada tahap sebelumnya, komponen khusus untuk *Table View Planner* dan *Template* belum tersedia. Oleh karena itu, dilakukan pengembangan tambahan untuk dua komponen tersebut.

Fitur ini dibangun dengan memanfaatkan komponen yang sudah ada, yaitu `<TripPlannerItemTable>`. Meskipun demikian, struktur kolom pada *planner* dan *template* berbeda dari komponen tabel lainnya, sehingga dibutuhkan definisi kolom (*columns*) khusus untuk masing-masing komponen.





Gambar 3.34. Tampilan fitur *Table View*

### 3.5 Kendala dan Solusi yang Ditemukan

#### 3.5.1 Kendala

Selama proses pengembangan dashboard, beberapa kendala teknis dan fungsional ditemukan. Berikut adalah daftar kendala yang muncul:

1. Peletakkan komponen pada file. Beberapa komponen memiliki nama file yang sama, namun digunakan di lokasi atau konteks yang berbeda. Hal ini menyebabkan kebingungan dalam struktur folder dan kesalahan import pada saat pengembangan.
2. Penggunaan Firebase Functions untuk mengambil dan menyimpan data. Integrasi antara frontend dengan Firebase Functions sempat mengalami kesulitan, terutama dalam menangani autentikasi dan format data saat pengambilan atau penyimpanan.

3. Penggunaan library React PDF untuk ekspor ke PDF. Library React PDF memiliki keterbatasan dalam hal styling dan tidak mendukung elemen HTML secara langsung, sehingga menyulitkan dalam menyesuaikan tampilan dokumen PDF agar menyerupai tampilan di halaman.

### 3.5.2 Solusi

1. *Refactor* beberapa file yang memiliki kegunaan yang sama dan isi yang sama namun hanya saja menggunakan beberapa *case* seperti `isConsultation`, `isTemplate`, dan sebagainya untuk menyesuaikan kebutuhannya.
2. Dokumentasi Firebase dibaca ulang dan debugging dilakukan secara bertahap dengan menambahkan `console.log()` pada Firebase Functions. Selain itu, digunakan middleware untuk menangani autentikasi secara global sebelum memproses fungsi utama.
3. Mempelajari dan mengikuti beberapa dari komponen yang pernah dibuat sebelumnya dan dibuat ulang komponen yang ingin di ekspor ke pdf menggunakan komponen dari React PDF seperti `<Text>`, `<View>`, dan `<Page>`. Untuk konsistensi gaya, dilakukan konversi manual dari gaya HTML ke sistem styling inline React PDF. Selain itu, dilakukan pengujian untuk setiap template PDF secara terpisah agar memastikan hasil ekspor sesuai harapan.

