

BAB III

METODOLOGI PENELITIAN

Pada bab ini akan membahas tentang tahapan-tahapan apa saja dalam Pengembangan Aplikasi Catatan Keuangan Untuk Usaha Mikro Kecil Menengah (Umkm) Berbasis Flutter untuk membantu mempermudah pencatatan keuangan pengusaha UMKM dalam melakukan pencatatan keuangan mereka.

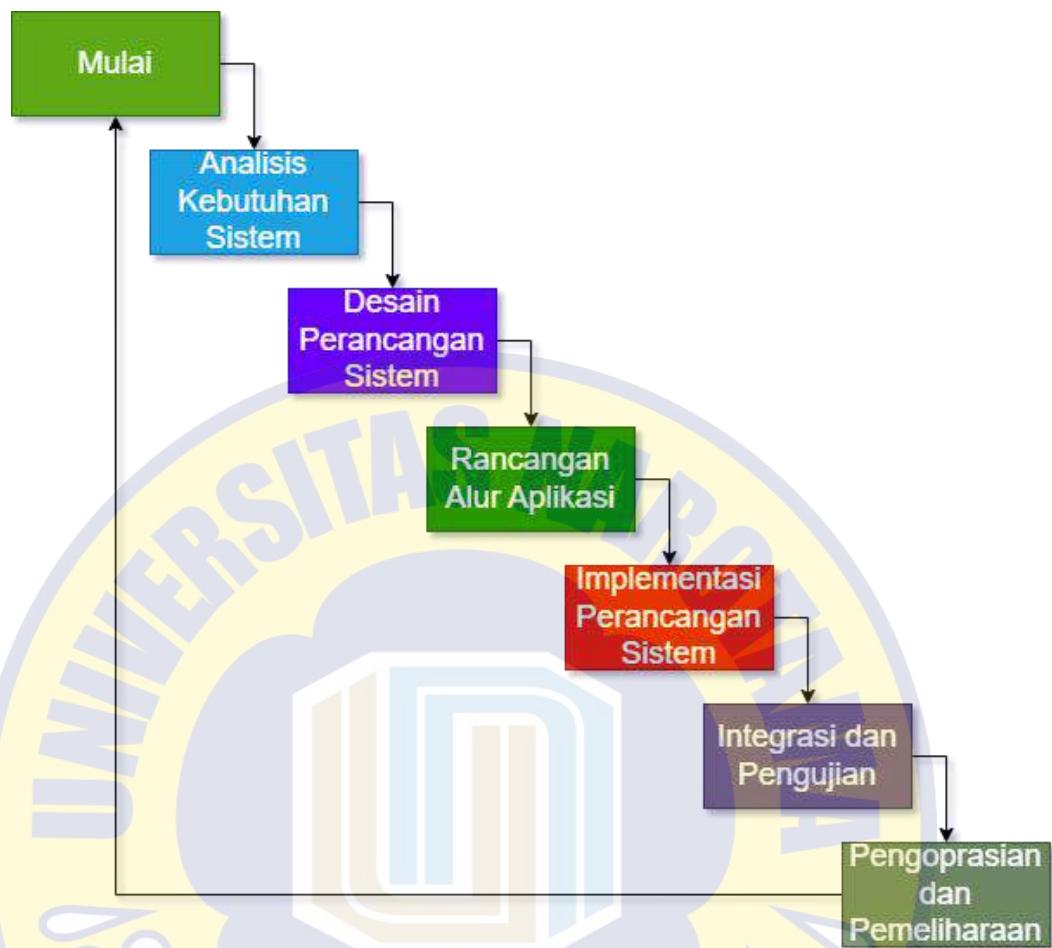
3.1 Pengembangan Ide

Pada saat melakukan observasi dan juga wawancara di lapangan, peneliti menemukan bahwa pengusaha Usaha Menegah Menengah Kecil (UMKM) masih menggunakan buku konfensional dan bahkan hanya menggunakan kertas dari sobekan bungkus kardus rokok untuk mencatat semua hasil keuangan mereka yang mana mereka memiliki kendala seperti kehilangan catatan tersebut, catatan yang memudar atau kehilangan lembaran tersebut. Sedangkan pada era teknologi saat ini yang di mana perkembangan teknologi sangat maju dan juga murah untuk di peroleh. Para pengusaha tersebut paling tidak memiliki satu *smartphone* dan tidak pernah lepas dari *smartphone* mereka.

Maka dari itu peneliti ingin mengembangkan aplikasi yang dapat digunakan pada perangkat genggam para pengusaha tersebut agar mereka dapat mempermudah dalam mencatat keuangan mereka dan juga menghindari adanya kehilangan data.

3.2 Metode Pengembangan Sistem

Dalam pengembangan aplikasi. Metode yang di gunakan selama pengembangan aplikasi Buku Kas Kita ini adalah dengan menggunakan model *waterfall*. Pada artikel eduNitas, model *waterfall* adalah proses desain berurutan, sering digunakan dalam proses pengembangan perangkat lunak, di mana kemajuan dilihat mengalir terus ke bawah (seperti air terjun) melalui fase Konsepsi, Inisiasi, Analisis, Desain, Konstruksi, Pengujian, Produksi/Implementasi, dan pemeliharaan(Model, 2015). Adapun tahap-tahap yang dilakukan dalam pengembangan aplikasi Buku Kas Kita, yaitu :



Gambar 3. 1 Perancangan Aplikasi Buku Kas Kita dengan menggunakan model waterfall

3.2.1 Analisis Kebutuhan Sistem

Dalam tahapan ini pengumpulan data yang di perlukan dalam melaksanakan penelitian ini peneliti menggunakan metode observasi dan wawancara di lapangan yang bertujuan untuk mengidentifikasi dan mengevaluasi permasalahan, peluang, dan hambatan yang terjadi serta kebutuhan yang di harapkan, sehingga dapat diusulkan perbaikan. Berikut adalah informasi yang peneliti dapatkan pada saat melakukan observasi dan wawancara :

tabel 3. 1 Tabel hasil observasi dan wawancara

Tanggal Pelaksanaan Wawancara dan Observasi	Narasumber	Pekerjaan	Permasalahan
Senin, 30 September 2024	Bapak Tri Effendy	Pedagang Sayur	Pada saat melakukan pencatatan pada lembar sobekan bungkus rokok sering terjadi kehilangan data, sedangkan bapak tersebut mempunyai sebuah <i>smartphone</i> tetapi tidak dapat digunakan untuk mencatat data keuangan

Pada permasalahan yang di dapatkan pada saat observasi dan wawancara di lapangan dapat di simpulkan :

Tabel 3. 2 Tabel Permasalahan dan Solusi

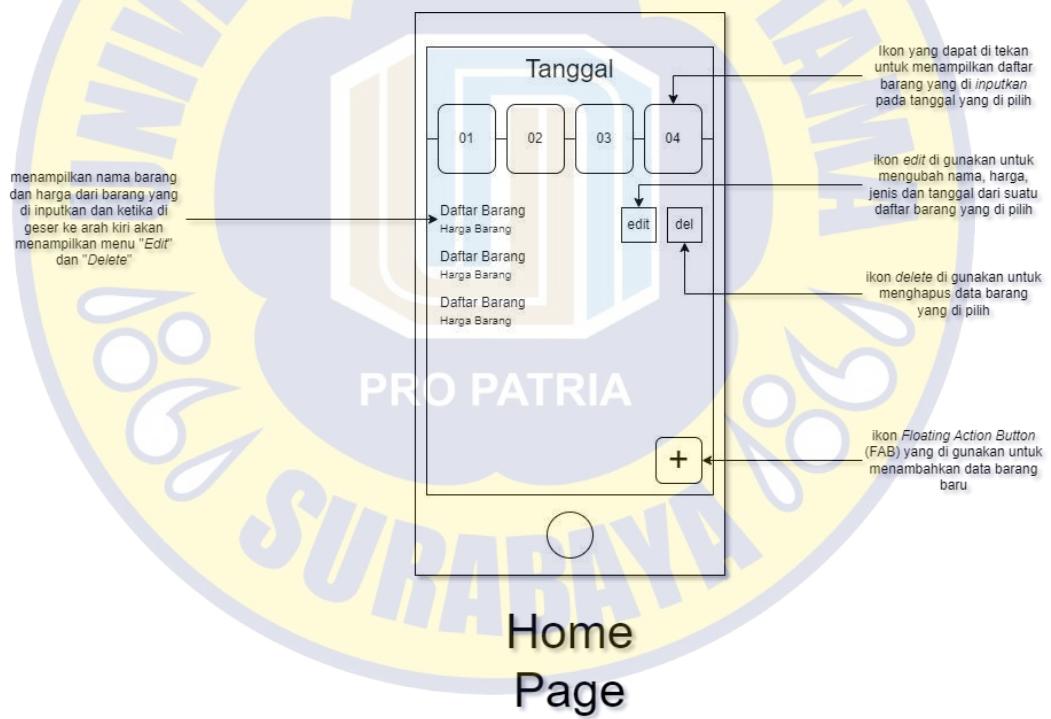
Permasalahan yang di dapat	Solusi yang di ajukan
Pada saat melakukan pencatatan sering kali terjadi kehilangan data, sedangkan para pelaku UMKM tersebut memiliki	Membuat sebuah aplikasi <i>mobile</i> yang dapat di gunakan untuk mencatat

setidaknya satu buah <i>smartphone</i> yang sering digunakan dan sering dibawa	semua catatan keuangan para pelaku UMKM
--	---

3.2.2 Desain Perancangan Sistem

Setelah mengetahui permasalahan yang di hadapi pelaku bisnis UMKM dan juga solusi yang di ajukan. Pada tahap ini peneliti menganalisis rancangan sebuah aplikasi *mobile* yang diharapkan dapat digunakan dan mempermudah pencatatan keuangan para pelaku UMKM. Berikut adalah rancangan umum dari aplikasi yang akan di bangun aplikasi :

A. Halaman *Home Page*



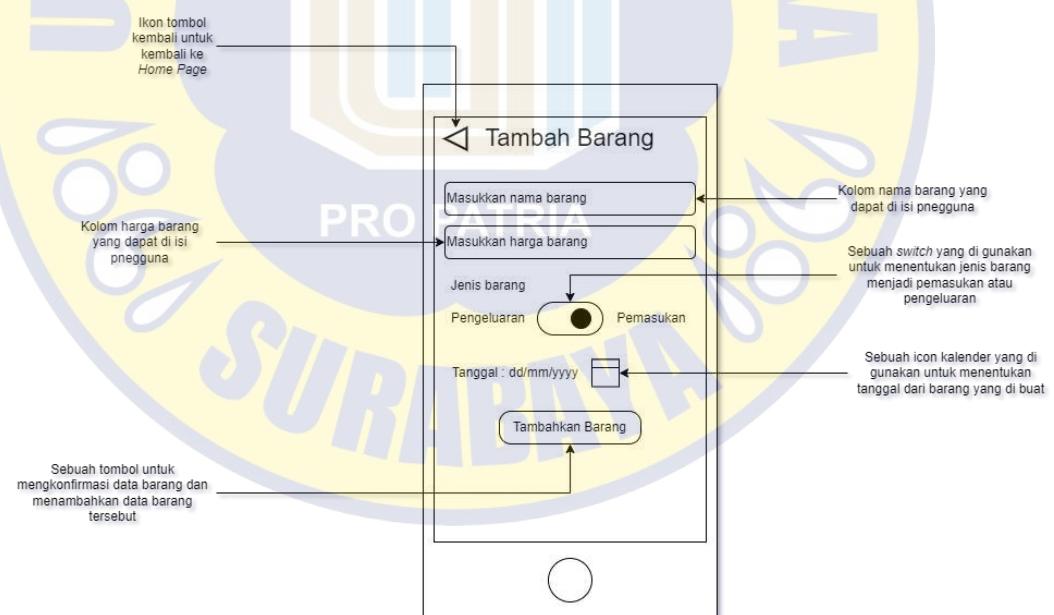
Gambar 3. 2 Rancangan tampilan dari menu *Home Page*

Penjelasan pada gambar 3.2 adalah sebagai berikut :

Pada saat pengguna membuka aplikasi, maka aplikasi akan menampilkan menu dari *Home Page*. Yang mana berisikan beberapa ikon yang berfungsi sebagai berikut :

- Ikon pada *appbar* di gunakan untuk menampilkan baris tanggal pada bulan tertentu, dan ketika pengguna menekan salah satu baris tanggal maka aplikasi akan menampilkan daftar barang yang ada pada tanggal tersebut.
- Tampilan baris daftar barang di gunakan untuk menampilkan nama, harga dan tanggal dibuat suatu barang yang di inputkan.
- Ikon *edit* digunakan untuk memindahkan pengguna ke halaman baru untuk mengubah suatu data barang yang di pilih
- Ikon *delete* digunakan untuk menghapus data barang yang di pilih oleh pengguna.
- Ikon *Floating Action Button* (FAB) digunakan untuk memindahkan tampilan pengguna ke halaman lain untuk menambahkan data barang baru.

B. Halaman Tambah Barang



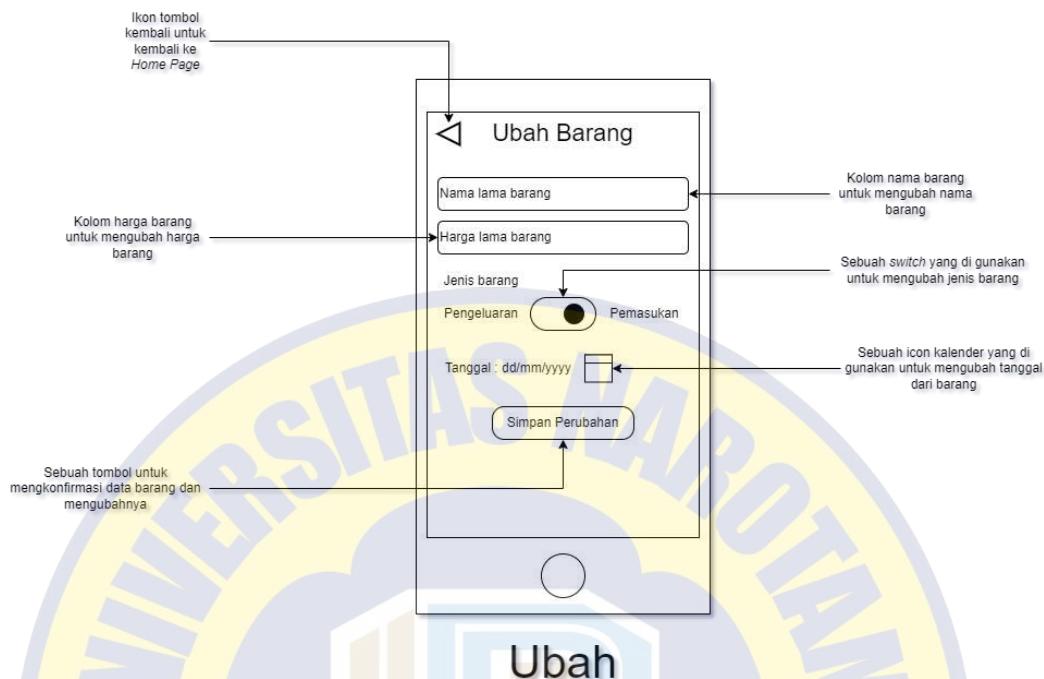
**Tambah
Barang
Page**

Gambar 3. 3 Rancangan tampilan menu Tambah Barang

Pada gambar 3.3 Pada halaman ini pada saat pengguna menekan Floating Action Button pada *Home Page*, pengguna akan di arahkan ke halaman menu Tambah Barang. Pada halaman Tabah Barang pengguna akan diminta untuk mengisi beberapa inputan yang disediakan. Beberapa inputan tersebut adalah sebagai berikut :

- Ikon tombol kembali digunakan untuk mengarahkan kembali pengguna ke halaman utamaho *Home Page*.
- Kolom nama barang digunakan untuk meminta pengguna mengisi nama barang yang di inginkan ke dalam kolom agar dapat di tambahkan ke dalam daftar barang.
- Kolom harga barang digunakan untuk meminta pengguna mengisi harga barang yang di inginkan ke dalam kolom agar dapat di tambahkan ke dalam daftar barang.
- Sebuah *Switch* digunakan untuk menetapkan jenis barang yang akan di tambahkan berupa pemasukan atau pengeluaran.
- Ikon tanggal digunakan untuk menetapkan tanggal berapa barang tersebut ingin di simpan agar dapat ditampilkan pada saat memilih tanggal pada halaman *Home Page*.
- Sebuah tombol “Tambah Barang” digunakan untuk menyimpan semua inputan yang telah di masukkan oleh pengguna ke dalam database.

C. Halaman Edit Barang



Gambar 3. 4 Rancangan tampilan menu Ubah Barang

Penjelasan pada gambar 3.4 adalah sebagai berikut :

Pada halaman ini pada saat pengguna menekan ikon edit pada *Home Page*, pengguna akan diarahkan ke halaman menu Ubah Barang. Pada halaman Ubah Barang pengguna akan diminta untuk mengisi beberapa inputan yang disediakan untuk di ubah. Beberapa inputan tersebut adalah sebagai berikut :

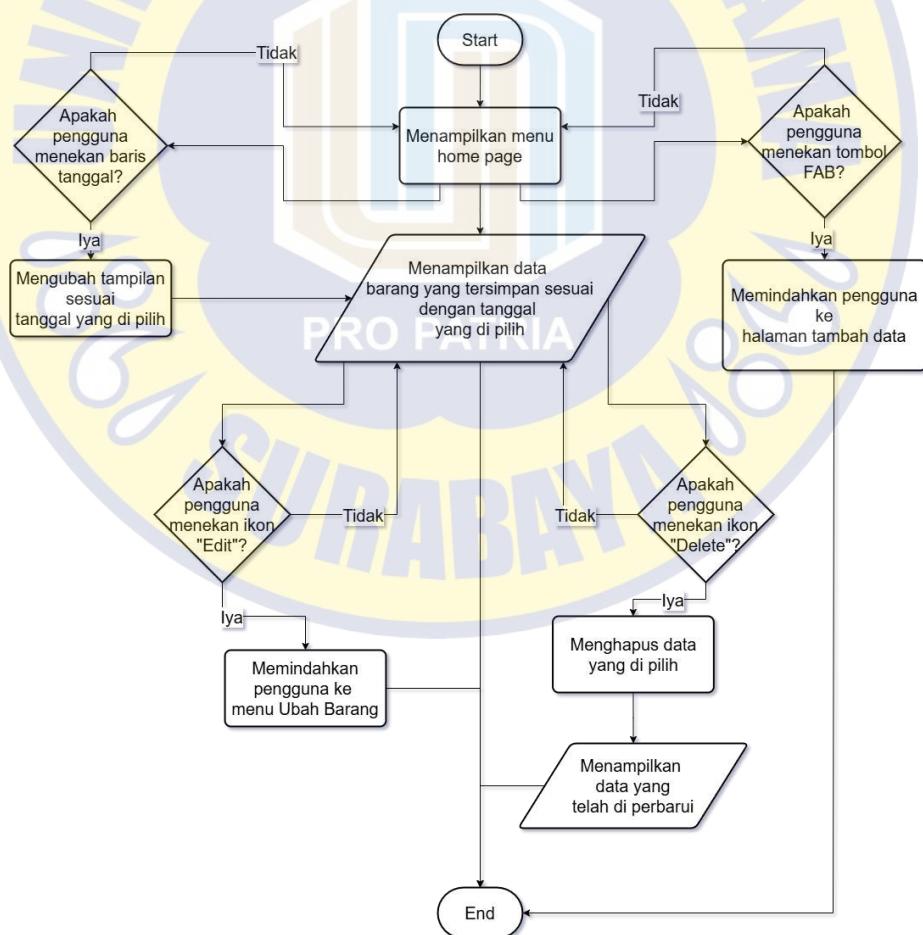
- Ikon tombol kembal digunakan untuk mengarahkan kembali pengguna ke halaman utama *Home Page*.
- Kolom nama barang digunakan untuk meminta pengguna mengisi nama barang yang di inginkan untuk dapat di ubah pada daftar barang.
- Kolom harga barang digunakan untuk meminta pengguna mengisi harga barang yang di inginkan untuk dapat di ubah pada daftar barang.
- Sebuah *Switch* digunakan untuk menetapkan perubahan jenis barang yang akan di ubah berupa pemasukan atau pengeluaran.

- Ikon tanggal digunakan untuk mengubah tanggal barang tersebut agar dapat ditampilkan pada saat memilih tanggal pada halaman *Home Page*.
- Sebuah tombol “Simpan Perubahan” digunakan untuk menyimpan semua inputan perubahan yang telah di masukkan oleh pengguna ke dalam database.

3.2.3 Rancangan Alur Aplikasi

Dalam merancang aplikasi, peneliti menggunakan diagram alir untuk membantu memvisualisasikan bagaimana program akan berjalan, mulai dari awal hingga akhir. Ini memungkinkan untuk melihat secara keseluruhan proses bagaimana aplikasi berjalan.

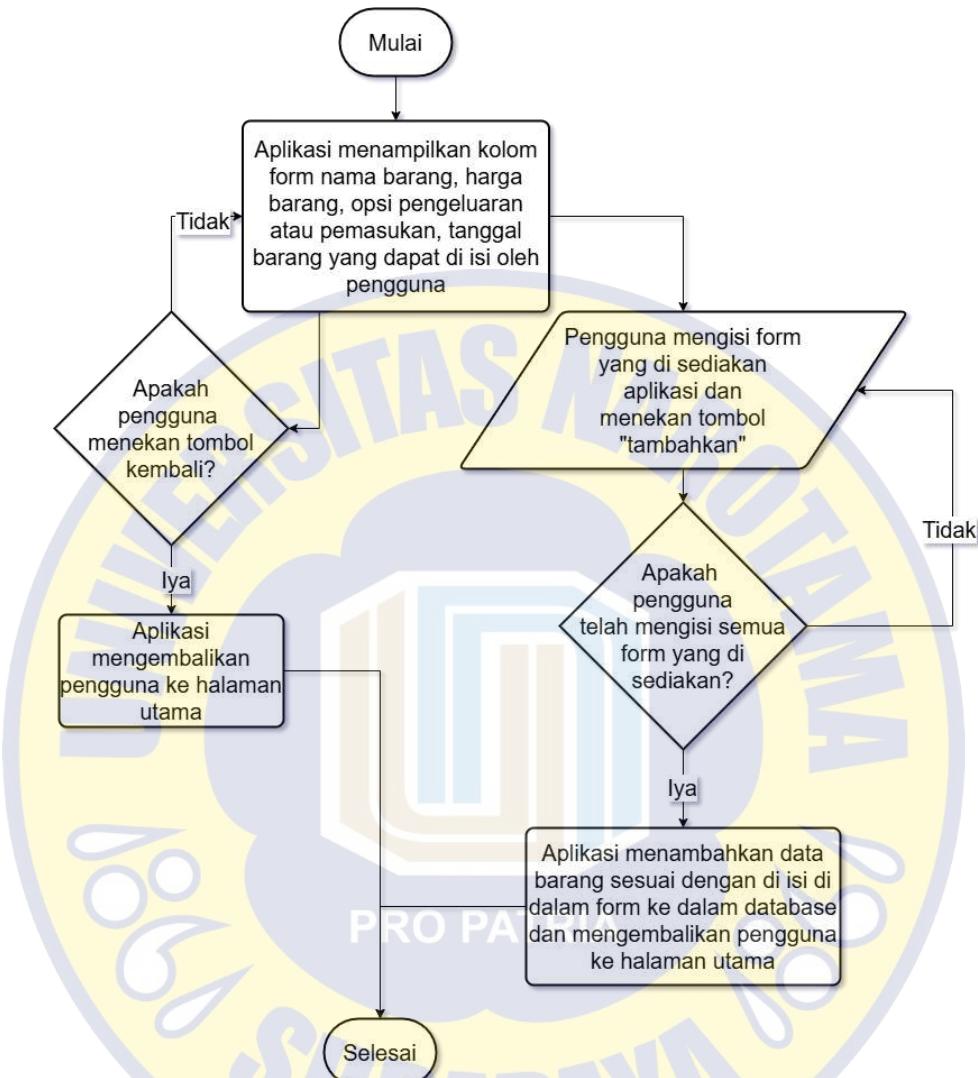
A. Alur Sistem Dalam Halaman Home Page



Gambar 3.5 Diagram alir dari halaman *Home Page*

Pada gambar 3.5 adalah diagram alir dari aplikasi yang akan dikembangkan, pada bagian awal setelah pengguna membuka aplikasi, aplikasi akan menampilkan halaman dari *Home Page* dan juga menampilkan data catatan yang tersimpan jika ada pada hari tersebut. Pada halaman ini jika pengguna memilih salah satu ikon tanggal aplikasi akan menampilkan data catatan pada tanggal yang dipilih oleh pengguna jika pengguna tidak menekan pilihan tanggal maka aplikasi tidak akan melakukan eksekusi. Adakalanya pengguna ingin menambahkan data catatan baru maka dari itu jika pengguna menekan tombol *Floating Action Button* (FAB) maka aplikasi tersebut akan mengarahkan pengguna ke halaman Tambah Barang dan jika pengguna tidak menekan tombol *Floating Action Button* (FAB) maka aplikasi tidak akan melakukan eksekusi untuk memindahkan pengguna ke halaman Tambah Barang. Pada tampilan data barang, jika pengguna ingin mengubah data dari barang yang diinginkan, maka ketika pengguna menekan tombol “*edit*” aplikasi akan mengarahkan pengguna ke halaman Ubah Barang, dan jika pengguna tidak menekan tombol “*edit*” maka aplikasi tidak akan melakukan eksekusi untuk memindahkan pengguna ke halaman Ubah Barang. Hampir sama dengan penggunaan tombol “*edit*”, jika pengguna ingin menghapus data barang yang dipilih oleh pengguna, maka pengguna dapat menekan tombol “*delete*” untuk menghapus data yang dipilih oleh pengguna, dan jika pengguna tidak menekan tombol “*delete*” maka aplikasi tidak akan melakukan eksekusi untuk menghapus data.

B. Alur Pada Halaman Tambah Barang

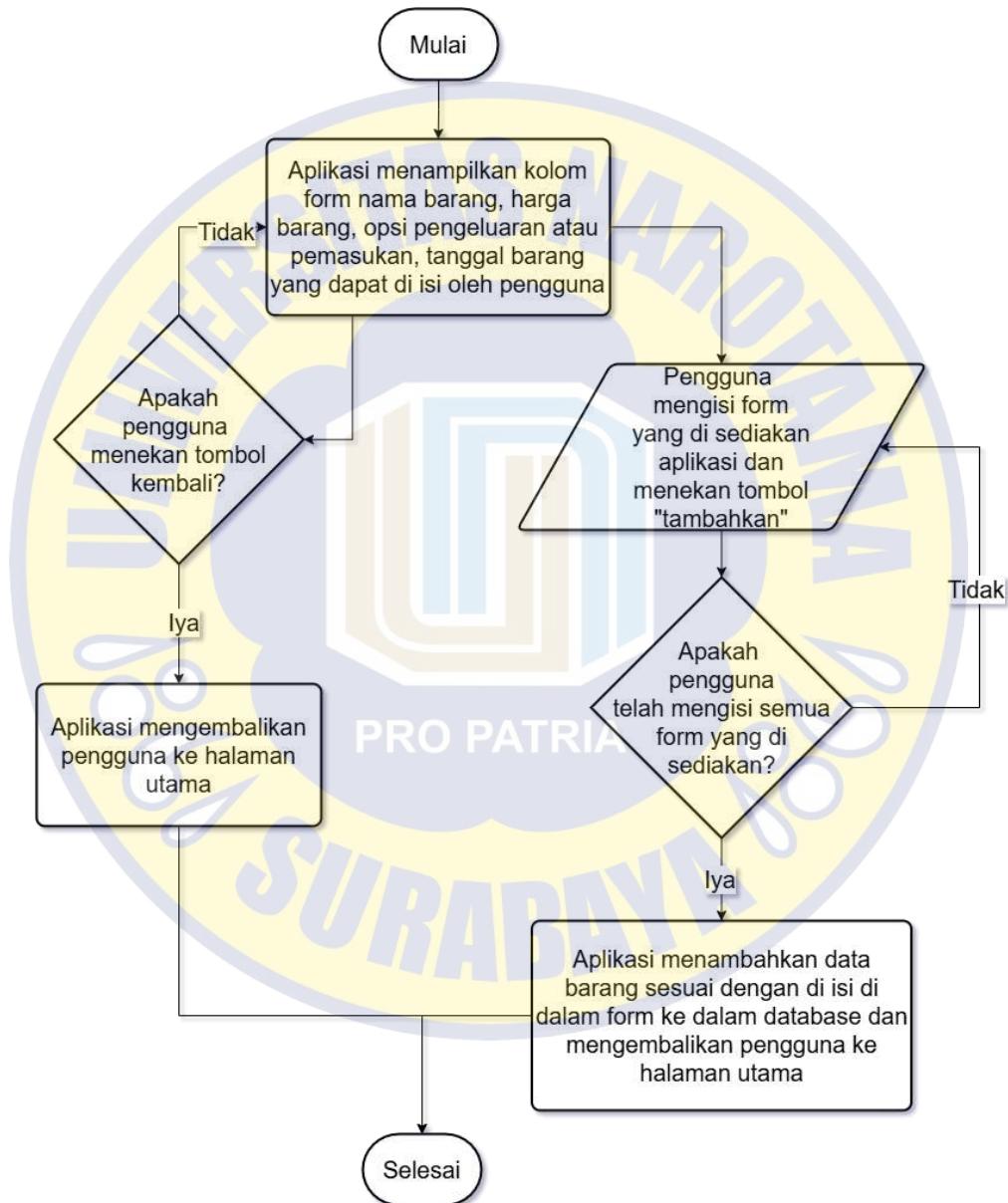


Gambar 3.6 Diagram alir dari halaman Tambah Barang

Pada gambar 3.6 di atas adalah diagram alir pada menu tambah barang yang mana pada halaman utama ketika pengguna menekan tombol *Floating Action Button* (FAB) maka pengguna akan diarahkan ke halaman tambah barang. Pada menu ini pada bagian *appbar* aplikasi terdapat sebuah tombol untuk menavigasi ke halaman utama. Pada bagian *body* aplikasi terdapat dua kolom yang harus diisi oleh pengguna lalu ada sebuah *switch* atau saklar untuk memberikan jenis dari barang tersebut berupa pengeluaran atau pemasukan, dan juga terdapat sebuah tombol untuk menentukan tanggal dari barang tersebut. Setelah semua kolom isi,

switch untuk menentukan jenis barang dan menentukan dari tanggal barang tersebut di masukkan, pengguna dapat menekan tombol “Tambahkan Barang” untuk menambahkan data barang baru tersebut ke dalam *database* aplikasi.

C. Alur Pada Menu Ubah Barang



Gambar 3. 7 Diagram alir dari halaman Ubah Barang

Pada gambar 3.7 di atas adalah diagram alir pada menu tambah barang yang mana pada halaman utama ketika pengguna menggeser dan menekan tombol pada

salah satu barang, maka pengguna akan diarahkan ke halaman Ubah Barang. Pada menu ini pada bagian *appbar* aplikasi terdapat sebuah tombol untuk menavigasi ke halaman utama. Pada bagian *body* aplikasi terdapat dua kolom yang harus diisi oleh pengguna lalu ada sebuah *switch* atau saklar untuk memberikan jenis dari barang tersebut berupa pengeluaran atau pemasukan, dan juga terdapat sebuah tombol untuk menentukan tanggal dari barang tersebut. Setelah semua kolom isi, *switch* untuk menentukan jenis barang dan menentukan dari tanggal barang tersebut di masukkan, pengguna dapat menekan tombol “Ubah” untuk mengubah data barang tersebut.

3.2.4 Implementasi Rancangan Sistem

Tahap ini dilakukan untuk mengimplementasikan dari tahap yang sebelumnya yaitu desain perancangan sistem. Di mana seluruh desain perancangan sistem yang telah dibuat sebelumnya diimplementasikan ke dalam bentuk kode-kode program sehingga dapat menghasilkan sebuah modul-modul yang akan diintegrasikan menjadi sistem yang lengkap.

3.2.5 Integrasi dan Pengujian

Setelah dilakukan tahapan implementasi dan menghasilkan modul-modul yang ada, maka ditahap ini dilakukan penggabungan modul-modul yang sebelumnya telah dibuat sehingga dapat menjadi sebuah sistem yang lengkap. Yang kemudian dilakukan juga pengujian apakah sistem yang dibuat desain dan fungsinya telah sesuai.

3.2.6 Pengoperasian dan Pemeliharaan

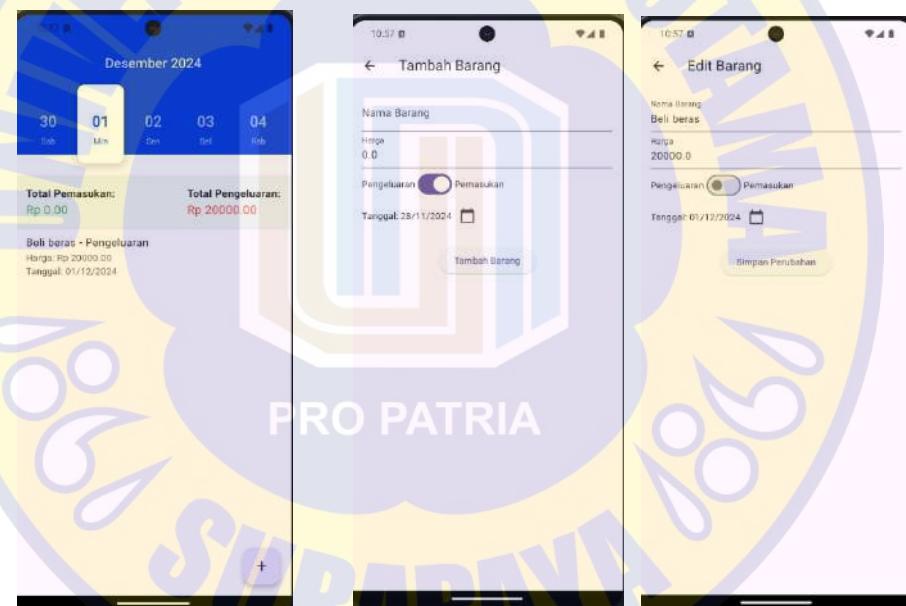
Tahapan ini merupakan tahap terakhir dari pembuatan aplikasi Buku Kas Kita yaitu pemeliharaan aplikasi sistem. Sistem yang telah berjalan perlu adanya pemeliharaan berguna untuk menyesuaikan dengan perkembangan yang harus dilakukan agar tidak usang.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan *User Interface* (UI)

Pada tahap ini adalah tahap pertama dalam pengembangan aplikasi, di mana peneliti membuat desain untuk aplikasi yang akan dibuat. Tahap ini diperlukan untuk menentukan fitur-fitur apa saja yang akan ada di dalam aplikasi yang akan dibuat yang nantinya setelah membuat *user interface* (UI) yang sesuai maka tahap selanjutnya adalah membuat interaksi antara fitur-fitur yang sudah dibuat agar aplikasi dapat berjalan sesuai yang diinginkan.



Gambar 4.1 Tampilan *User Interface* (UI) *Home screen*, *Tambah Barang*, dan *Ubah Barang* pada aplikasi yang dibuat

Pada gambar 4.1 merupakan tampilan dari *Home Screen*, *Tambah Barang*, dan *Ubah Barang* pada aplikasi yang dikembangkan. Pada desain tersebut diharapkan dapat menyampaikan tujuan peneliti dalam mempermudah dan menambah kepuasan pengguna.

4.2 Implementasi Halaman Home Page

Pada tahap ini peneliti mengimplementasikan desain dari *Home Page* yang telah dibuat ke dalam baris-baris kode *Dart* dengan framework *Flutter*.

```
20 ...Widget·build(BuildContext·context){  
21 ...    return·Scaffold(  
22 ...        appBar::CalendarAppBar(  
23 ...            onDateChanged::(date){  
24 ...                setState((){  
25 ...                    selectedDate:=date;  
26 ...                });  
27 ...            },  
28 ...            backButton::false,  
29 ...            locale::"id",  
30 ...            firstDate::DateTime.now().subtract(const·Duration(days:-365)),  
31 ...            lastDate::DateTime.now().add(const·Duration(days:10)),  
32 ...        ), // CalendarAppBar
```

Gambar 4.2 implementasi *package* *Calendarappbar* pada *home screen* aplikasi

Pada gambar 4.2 adalah baris kode penggunaan *package* *CalendarAppbar* yang digunakan untuk membuat baris tanggal pada bagian atas aplikasi pada menu *home screen* yang di mana pengguna dapat menentukan dan memilih tanggal yang di inginkan untuk melihat barang apa saja yang telah dibuat pada tanggal yang dipilih.

```
93  
94 ...    return·ListView.builder(  
95 ...        padding::const·EdgeInsets.all(16),  
96 ...        itemCount::filteredItems.length,  
97 ...        itemBuilder::(context, index){  
98 ...            final·item:=filteredItems[index];  
99 ...            return·Card(  
100 ...                elevation::3,  
101 ...                margin::const·EdgeInsets.symmetric(vertical:6),  
102 ...                child::ListTile(  
103 ...                    title::Text(  
104 ...                        '${item.name}·-${item.isIncome?·'Pemasukan':·'Pengeluaran'}', // Text  
105 ...                        subtitle::Text(  
106 ...                            'Harga: ${NumberFormat.currency(locale:'id', symbol:'Rp').format(item.price)}\n'  
107 ...                            'Tanggal: ${DateFormat('dd/MM/yyyy').format(item.createdAt)}',  
108 ...                        )  
109 ...                )  
110 ...            );  
111 ...        }  
112 ...    );  
113 ...}
```

Gambar 4.3 Baris kode daftar barang pada bagian *home page* aplikasi

Pada gambar 4.3 merupakan kumpulan kode-kode dalam menampilkan jumlah pengeluaran dan pemasukan barang dan juga untuk menampilkan urutan barang yang ada pada tanggal yang di pilih pada menu *home screen*.

```
65 .....items: const [
66 .....  BottomNavigationBarItem(
67 .....    icon: Icon(Icons.list),
68 .....    label: "Transaksi",
69 .....  ), // BottomNavigationBarItem
70 .....  BottomNavigationBarItem(
71 .....    icon: Icon(Icons.pie_chart),
72 .....    label: "Ringkasan",
73 .....  ), // BottomNavigationBarItem
74 .....],
75 .....), // BottomNavigationBar
```

Gambar 4.4 Baris kode untuk menampilkan menu tab bar

Pada gambar 4.4 merupakan baris kode untuk menampilkan ikon menu transaksi barang dan juga untuk menampilkan ringkasan keuangan pada bagian tab bar yang akan di tampilkan pada bagian bawah aplikasi.

```
193 .....  Text(
194 .....    "Ringkasan Bulanan ${DateFormat('MMMM yyyy').format(selectedDate ?? DateTime.
195 .....      now())}",
196 .....    style: const TextStyle(
197 .....      fontSize: 18, fontWeight: FontWeight.bold), // TextStyle
198 .....  const SizedBox(
199 .....    height: 15,
200 .....  ), // SizedBox
201 .....  _buildSummaryTile("Pemasukan Bulanan",
202 .....    totalPemasukanBulanan, Colors.green),
203 .....  _buildSummaryTile("Pengeluaran Bulanan",
204 .....    totalPengeluaranBulanan, Colors.red),
205 .....  const Divider(),
206 .....  _buildSummaryTile("Total Bulanan", totalBulanan,
```

Gambar 4.5 Baris kode untuk menampilkan ringkasan bulanan

Pada gambar 4.5 merupakan baris kode untuk menampilkan ringkasan bulanan mulai dari total pemasukan, total pengeluaran, dan total keseluruhan yakni

pengurangan total pemasukan yang di kurangi dengan total pengeluaran bulanan pada menu ringkasan.

```
230 .....      Text(.....  
231 .....      "Ringkasan·Harian·(${DateFormat('dd·MMM·yyyy').format(selectedDate·??·DateTime.  
now())})",  
232 .....      style:·const·TextStyle(  
233 .....      ....fontSize:·18,·fontWeight:·FontWeight.bold), // TextStyle  
234 .....      ), // Text  
235 .....      const·SizedBox(  
236 .....      ....height:·15,  
237 .....      ), // SizedBox  
238 .....      _buildSummaryTile(  
239 .....      ...."Pemasukan·Harian",·totalPemasukanHarian,·Colors.green),  
240 .....      _buildSummaryTile("Pengeluaran·Harian",  
241 .....      ....totalPengeluaranHarian,·Colors.red),  
242 .....      const·Divider(),  
243 .....      buildSummaryTile("Total·Harian",·totalHarian,
```

Gambar 4.6 Baris kode untuk menampilkan ringkasan harian

Pada gambar 4.6 merupakan baris kode untuk menampilkan ringkasan harian mulai dari total pemasukan, total pengeluaran, dan total keseluruhan yakni pengurangan total pemasukan yang di kurangi dengan total pengeluaran harian pada menu ringkasan.

```
42 .....      floatingActionButton:·FloatingActionButton(  
43 .....      ....onPressed:·()·{  
44 .....      ....Navigator.push(  
45 .....      ....context,  
46 .....      ....MaterialPageRoute(builder:·(_)...=>·const·ItemForm()),  
47 .....      ....);  
48 .....      },  
+ 49 .....      child:·const·Icon(Icons.add),
```

Gambar 4.7 Baris kode untuk menuju menu Tambah Barang

Pada gambar 4.7 merupakan baris kode untuk menampilkan sebuah tombol *Floating Action Button* (FAB) pada bagian bawah tengah aplikasi yang di gunakan ketika pengguna ingin menambahkan data barang baru maka pengguna tinggal menekan tombol tersebut maka aplikasi akan mengarahkan pengguna menuju halaman Tambah Barang.

```
112 ..... IconButton(  
113 ..... icon: const Icon(Icons.edit),  
114 ..... onPressed: () => Navigator.push(  
115 ..... context,  
116 ..... MaterialPageRoute(  
117 ..... builder: (_) => ItemForm(item: item),  
118 ..... ), // MaterialPageRoute  
119 ..... ),  
120 ..... ), // IconButton
```

Gambar 4.8 Baris kode untuk menuju halaman Ubah Barang

Pada gambar 4.8 merupakan baris kode yang di mana ketika pengguna ingin mengubah data dari suatu barang maka pengguna dapat menekan gambar pensil di sebelah data barang yang di pilih, maka pengguna hanya tinggal menekan gambar tersebut maka aplikasi akan memindahkan pengguna ke halaman Ubah Barang.

```
121 ..... IconButton(  
122 ..... icon: const Icon(Icons.delete, color: Colors.red),  
123 ..... onPressed: () => {  
124 ..... box.delete(item.key);  
125 ..... },  
126 ..... ), // IconButton
```

Gambar 4.9 Baris kode untuk menghapus data barang

Pada gambar 4.9 merupakan baris kode yang di mana ketika pengguna ingin menghapus data barang yang sudah tidak dinginkan lagi, maka pengguna hanya perlu menekan gambar keranjang sampah pada bagian kanan data barang yang di pilih, maka secara otomatis aplikasi akan menghapus data barang tersebut.

4.3 Implementasi Halaman Tambah barang

Pada tahap ini peneliti mengimplementasikan desain dari Tambah Barang yang telah dibuat ke dalam baris-baris kode Dart dengan framework Flutter.

```
49     TextFormField(  
50       initialValue: _name,  
51       decoration: InputDecoration(labelText: 'Nama Barang'), Use 'c'  
52       validator: (value) {  
53         if (value == null || value.isEmpty) {  
54           return 'Nama tidak boleh kosong';  
55         }  
56         return null;  
57       },  
58       onSaved: (value) {  
59         _name = value!;  
60       },  
61     ), // TextFormField
```

Gambar 4.10 Baris kode untuk membuat komponen *form* nama barang

Pada gambar 4.10 kode tersebut di gunakan untuk membuat sebuah kolom *form* untuk nama barang yang digunakan untuk menentukan nama dari barang yang di inginkan pengguna.

```
62     TextFormField(  
63       initialValue: _price.toInt().toString(),  
64       decoration: InputDecoration(labelText: 'Harga'), Use 'const'  
65       keyboardType: TextInputType.number,  
66       validator: (value) {  
67         if (value == null || value.isEmpty) {  
68           return 'Harga tidak boleh kosong';  
69         }  
70         return null;  
71     },  
72     onSaved: (value) {  
73       _price = double.parse(value!);  
74     },  
75   ), // TextFormField
```

Gambar 4.11 Baris kode untuk membuat kolom *form* harga barang

Pada gambar 4.11 merupakan baris kode yang di gunakan untuk membuat kolom *form* harga barang yang di gunakan untuk menentukan harga dari barang yang pengguna inginkan.

```
82     Text('Pengeluaran'),      Use 'const' with the constructor to im
83     Switch(
84       value:_isIncome,
85       onChanged:(value){
86         setState((){
87           _isIncome=_value;
88         });
89       },
90     ), // Switch
91     Text('Pemasukan'),      Use 'const' with the constructor to impr
92   ],
93 ), // Row
```

Gambar 4.12 Baris kode *switch* untuk menentukan jenis barang

Pada gambar 4.12 merupakan baris kode untuk menampilkan saklar atau switch yang di gunakan untuk menentukan jenis dari barang yang ingin di *inputkan* oleh pengguna, jenis dari barang tersebut adalah yang jika saklar menyala maka jenis dari barang adalah pemasukan sedangkan jika saklar mati merupakan jenis pengeluaran.

```
97     Text(
98       'Tanggal: ${DateFormat('dd/MM/yyyy').format(_selectedDate)}'), // Text
99     IconButton(
100       icon: Icon(Icons.calendar_today),      Use 'const' with the constructor to :
101       onPressed: () async {
102         final selected=await showDatePicker(
103           context: context,
104           initialDate: _selectedDate,
105           firstDate: DateTime(2000),
106           lastDate: DateTime(2999),
107         );
108         if (selected!=null){
109           setState((){
110             _selectedDate=selected;
```

Gambar 4.13 Baris kode tanggal untuk menentukan tanggal dari barang

Pada gambar 4.13 merupakan baris kode untuk membuat sebuah ikon yang dapat di tekan oleh pengguna, yang di mana ketika pengguna menekan ikon tanggal tersebut maka pengguna dapat menentukan tanggal dari barang.

```
118 ElevatedButton(  
119   onPressed: () {  
120     if (_formKey.currentState!.validate()) {  
121       _formKey.currentState!.save();  
122       final box = Hive.box<Item>('items');  
123       if (widget.item != null) {  
124         widget.item!.name = _name;  
125         widget.item!.price = _price;  
126         widget.item!.isIncome = _isIncome;  
127         widget.item!.createdAt = _selectedDate;  
128         widget.item!.save();  
129       } else {  
130         final newItem = Item(  
131           name: _name,  
132           price: _price,
```

Gambar 4.14 Baris kode untuk mengkonfirmasi barang yang ingin di tambahkan

Pada gambar 4.14 merupakan beris kode untuk membuat tombol validasi yang digunakan untuk mengkonfirmasi data dari barang yang ingin di inputkan, yang di mana terdapat logika ketika pengguna masih belum melengkapi kolom *form* maka aplikasi akan menampilkan peringatan dan tidak menambahkan data tersebut. Sedangkan jika data yang di masukkan sudah lengkap maka aplikasi akan menambahkan data tersebut ke dalam *database*.

4.4 Implementasi Halaman Ubah Barang

Pada tahap ini peneliti menggunakan baris kode yang sama yang di gunakan pada Implementasi Halaman Tambah Barang, yang dikarenakan semua kebutuhan dalam mengubah data yang telah di *inputkan* oleh pengguna mulai dari nama barang, harga barang, jenis barang, dan tanggal barang dapat melalui baris kode yang sama. Maka dari itu peneliti menggunakan baris kode tersebut kembali.

```

7 ..@HiveField(0)
8 ..late String name;
9
10 ..@HiveField(1)
11 ..late double price;
12
13 ..@HiveField(2)
14 ..late bool isIncome;
15
16 ..@HiveField(3)
17 ..late DateTime createdAt;
18
19 ..Item(
20 .....{required this.name,
21 .....required this.price,
22 .....required this.isIncome,

```

Gambar 4.15 Model dari *database* aplikasi menggunakan *package Hive*

Pada gambar 4.5 merupakan baris kode untuk model database aplikasi yang dikembangkan. Pada baris tersebut berisi ketentuan untuk membuat *database* pada *package Hive* yang nantinya *package* tersebut dapat membuat *database* untuk aplikasi yang dikembangkan secara otomatis.

```

1 //·GENERATED·CODE·--·DO·NOT·MODIFY·BY·HAND
2
3 part·of·'item.dart';
4
5 //*****
6 //·TypeAdapterGenerator
7 //*****
8
9 class ItemAdapter extends TypeAdapter<Item>{
10 ..@override
11 ..final int typeId=0;
12
13 ..@override
14 ..Item read(BinaryReader reader){
15 .....final numFields=reader.readByte();
16 .....final fields=<int, dynamic>{
17 .....for (int i=0; i<numFields; i++) reader.readByte(): reader.read(),

```

Gambar 4. 16 hasil database yang di buat secara otomatis oleh package Hive

Pada gambar 4.6 merupakan hasil *database* dari *package hive* yang di mana baris kode tersebut di buat secara otomatis sesuai dengan model *database* sebelumnya. Hasil dari pembuatan *database* ini tidak boleh di ubah dikarenakan dapat merusak struktur dari *database* yang sudah di buat.

4.5 Uji Coba

Pada tahap ini, aplikasi yang sudah di buat akan di implementasikan dan akan di uji kepada pengguna yang di mana pengguna adalah seorang pedagang sayur keliling di daerah perumahan di daerah Krian. Selama pengujian ini berlangsung peneliti akan memantau kinerja dari aplikasi yang di buat untuk memastikan apakah aplikasi yang di buat dapat berjalan dengan baik atau tidak.



Gambar 4.17 Pelaku Usaha UMKM Pedagang Keliling

Pada saat pengujian aplikasi, peneliti berkesempatan untuk menguji aplikasi yang sudah di buat pada seorang pelaku usaha UMKM yakni seorang pedagang keliling di daerah Krian. Selama pengujian berlangsung peneliti berharap mendapatkan hasil berupa hasil kerja dari aplikasi yang sudah di buat apakah aplikasi tersebut bekerja dengan baik atau akan mengalami kendala pada saat pengujian berlangsung.

4.6 Hasil Pengujian Aplikasi

Setelah peneliti melakukan pengujian terhadap aplikasi yang telah dikembangkan, maka pada tahap ini peneliti ingin melihat kinerja dari aplikasi yang telah dikembangkan. Untuk mengetahui apakah fitur-fitur dalam aplikasi yang telah dikembangkan dapat bekerja dengan baik atau tidak, maka dari itu peneliti menggunakan metode Black Box dalam mengetahui kinerja fungsionalitas dari aplikasi yang dikembangkan. Berikut tabel fungsionalitas aplikasi yang telah dikembangkan :

Tabel 4.1 Tabel hasil pengujian fungsionalitas aplikasi

No	Fitur Yang Diuji	Fungsionalitas	
		Iya	Tidak
1	Apakah aplikasi dapat menampilkan daftar barang yang telah ditambahkan sesuai dengan tanggalnya?	✓	
2	Apakah tombol <i>Floating Action Button</i> (FAB) pada aplikasi dapat memindahkan pengguna ke halaman Tambah Barang?	✓	
3	Apakah aplikasi dapat menambahkan barang ke dalam <i>database</i> aplikasi?	✓	
4	Apakah aplikasi dapat menampilkan total pengeluaran, pemasukan, dan total keseluruhan untuk bulanan dan juga harianya?	✓	
5	Apakah tombol ubah barang pada setiap data barang dapat memindahkan pengguna ke halaman Ubah Barang?	✓	
6	Apakah aplikasi dapat mengubah data barang yang pengguna inginkan?	✓	
7	Apakah aplikasi dapat menghapus data barang yang dinginkan pengguna?	✓	

4.7 Hasil Uji Coba Pengguna

Selama melakukan uji coba dan observasi terhadap aplikasi yang telah dibuat, maka tahap selanjutnya adalah tahap hasil di mana peneliti ingin mengetahui kelayakan aplikasi yang dikembangkan apakah dapat berjalan dengan baik atau mengalami suatu kendala.

Setelah tahap pengujian dan observasi terhadap aplikasi yang telah dibuat, berikut adalah beberapa penjelasan hasil terkait kinerja dari aplikasi yang telah dibuat



Gambar 4.18 Pengujian Aplikasi

Setelah melakukan uji coba pada aplikasi yang telah dibuat, selama satu bulan pada awal hingga pertengahan pengujian. Aplikasi masih dapat berjalan sesuai dengan yang diharapkan yakni dapat menyimpan nama barang dan juga harga dari barang yang diinputkan ke dalam aplikasi.

Akan tetapi pada bulan terakhir pengujian kepada aplikasi terdapat sebuah masalah terjadi pada aplikasi, masalah yang muncul pada saat bulan terakhir pengujian aplikasi adalah setelah pengguna menambahkan barang ke dalam aplikasi, barang yang telah di tambahkan ke dalam aplikasi, barang tersebut tidak dapat di hapus oleh pengguna. Akan tetapi barang tersebut masih dapat di ubah baik nama barang maupun harga dari barang tersebut oleh pengguna, dan juga untuk fungsi menambahkan data barang beserta harga dari barang yang akan di inputkan oleh pengguna masih dapat berjalan dengan lancar.

Setelah pengujian dengan pengguna berakhir, peneliti meninjau aplikasi yang telah di kembangkan kembali untuk memperbaiki permasalahan yang ada. Setelah melakukan *debugging* pada aplikasi yang di kembangkan, peneliti menemukan permasalahan yang ada yakni kesalahan penulisan pada baris kode untuk menghapus data barang yang mana baris kode sebelumnya menghapus data barang pertama di dalam *database* aplikasi jika data barang yang ingin pengguna hapus berada pada urutan pertama pada tanggal yang di pilih, maka dari itu peneliti mengubah baris kode tersebut dengan membuat kunci untuk setiap data barang dan setelah peneliti memperbaiki permasalahan tersebut, aplikasi sekarang dapat berjalan dengan penuh kembali tanpa mengalami permasalahan yang berati.

BAB V

SARAN DAN KESIMPULAN

5.1 Kesimpulan

Pada penelitian ini yang berjudul PENGEMBANGAN APLIKASI CATATAN KEUANGAN UNTUK USAHA MIKRO KECIL MENENGAH (UMKM) BERBASIS FLUTTER. Kesimpulan yang di dapat selama menjalankan penelitian ini adalah sebagai berikut Selama awal pengujian aplikasi berlangsung, aplikasi masih berjalan dengan baik akan tetapi pada sesi akhir pengujian terdapat kendala pada aplikasi, yakni data yang di masukkan ke dalam aplikasi tidak dapat di hapus akan tetapi aplikasi masih dapat untuk menambahkan dan juga mengubah data yang telah di simpan. Tetapi setelah pengujian selesai peneliti berusaha dalam memperbaiki permasalahan yang ada dan akhirnya menemukan permasalahan tersebut dan memperbaikinya.

5.2 Saran

Saran yang di dapat selama melakukan penelitian terhadap aplikasi yang di buat berasal baik dari pengguna, dosen pembimbing, maupun dosen pengaji, berikut adalah saran yang di dapat selama penelitian berlangsung :

1. Saran dari pengguna agar aplikasi dapat menambah nama dan harga sekali lalu data tersebut dapat di pakai kembali pada hari berikutnya ketika ingin menambahkan data tersebut.
2. Saran dari dosen untuk menambahkan total keuangan pada setiap bulanannya

DAFTAR PUSTAKA

- Afandi, R. S., & Saputra, E. H. (2013). Aplikasi Mobile Informasi Kafe 24 Jam Di Yogyakarta Berbasis Android. *Data Manajemen Dan Teknologi Informasi (DASI)*, 14(3), 49.
- Al Ghani, A. I., & Andrian, R. (2023). Pengembangan Presensee: Aplikasi Presensi Mahasiswa Mobile Menggunakan Framework Flutter (Studi Kasus: Studi Independen Alterra Academy. *JURNAL MEDIA INFOTAMA*, 19(2), 447–453.
- Alfonso, G., & Setiawati, P. S. P. (2024). APLIKASI MANAJEMEN UANG KAS KELAS BERBASIS MOBILE DENGAN FRAMEWORK FLUTTER. *Journal of Computer Science and Information Technology*, 2(1), 15–25.
- ALICIA TUOVILA. (2023, October 17). *Apa Itu Buku Kas? Cara Kerja Buku Kas, Beserta Contohnya.*
- ANASTASIA, F. D. (2022). *Pengembangan Front-End Pada Aplikasi M-Banking Agen46 Dengan Teknologi Flutter.*
- Anastasia, F. D., & Paputungan, I. V. (2022). Implementasi Bloc Pattern Pada Pengembangan Frontend Fitur Top Up Linkaja Aplikasi M-Banking Agen46 Dengan Teknologi Flutter (Studi Kasus: PT. Bank Negara Indonesia Tbk). *Jurnal Sains, Nalar, Dan Aplikasi Teknologi Informasi*, 2(1), 1–11.
- Ardianto, M. F., Sopian, F. P., Irfan, M., & Harriansyah, H. (2024). Perancangan Aplikasi Bisnis Laundry Sepatu Berbasis Mobile Dengan Flutter. *Jurnal Maklumatika*, 94–105.
- Aziz, M. F., & Sanjaya, C. B. (2023). Aplikasi Kas Berbasis Flutter untuk Meningkatkan Efisiensi Pencatatan Transaksi Keuangan. *Jurnal Krisnadana*, 3(1), 34–48.
- Haniefam, H., Prasetijo, A. B., & Eridani, D. (2022). Perancangan Front End Aplikasi Reservasi Talanoa Kopi and Space Menggunakan Framework Flutter. *Jurnal Teknik Komputer*, 1(3), 120–125.
- Harahap, M. S., Andrianto, R., Dermilan, D., Elok, D., Khoiriah, C., & Andika, B. (2023). Pencatat Keuangan: Pengembangan Aplikasi Futter untuk Mencatat dan Mengelola Transaksi Keuangan. *Jurnal Penelitian Teknologi Informasi Dan Sains*, 1(3), 64–75.
- Hendriawan, M., Budiman, T., Yasin, V., & Rini, A. S. (2021). Pengembangan aplikasi e-commerce di pt. putra sumber abadi menggunakan flutter. *JISICOM (Journal of Information System, Informatics and Computing)*, 5(1), 69–88.
- Herlambang, R. R. A., & Zukhri, Z. (2023). Penggunaan Firebase Analytics pada Pengembangan Aplikasi Mobile I'm UII dengan Framework Flutter. *AUTOMATA*, 4(2).

- Irsan, M. (2015). Rancang bangun aplikasi mobile notifikasi berbasis android untuk mendukung kinerja di instansi pemerintahan. *JustIN (Jurnal Sistem Dan Teknologi Informasi)*, 3(1), 115–120.
- Model, W. (2015). Waterfall model. *Luettavissa: Http://Www. Waterfall-Model. Com/*. Luettu, 3.
- Prasetyo, G. W. T., Pradana, F., & Prakoso, B. S. (2022). Pengembangan Aplikasi Point of Sales Warung dan UMKM "WarunkQu" menggunakan Framework Flutter. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 6(10), 4724–4730.
- Ramadhan, S., & Julkarnain, M. (2024). MOBILE APLIKASI MANAJEMEN KEUANGAN CV. TOP DIGITAL PRINTING DOMPU. *Jurnal Teknologi Informatika Dan Komputer*, 2(1), 13–22.
- Rony Setiawan. (2021, November 17). *Black Box Testing Untuk Menguji Perangkat Lunak*. Dicoding.
- Sera, R. C. K. (2021). *Rancangan Bangun Aplikasi Money Changer Sederhana Berbasis Android*. Dec.
- Sudradjat, B. (2021). Penggunaan Teknologi Flutter dalam Aplikasi Mobile untuk Pengembangan Kedai Kopi. *REMIK: Riset Dan E-Jurnal Manajemen Informatika Komputer*, 6(1), 1–8.
- Wicaksono, A. P., Kharisma, A. P., & Sianturi, R. S. (2022). Pengembangan Aplikasi Mobile Manajemen Indekos berbasis Android dengan Metode Human-Centered Design (Studi Kasus: Kos Bu Parjo). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 6(8), 3596–3605.
- Wiguna, N. S., & Nanang, N. (2024). PERANCANGAN APLIKASI PENGGAJIAN KARYAWAN BERBASIS ANDROID DAN DART FLUTTER PADA PT ANDALAN K3 MENGGUNAKAN MODEL WATERFALL. *Jurnal Multidisipliner Bharasumba*, 3(04), 131–154.

LAMPIRAN

Lampiran 1 Hasil Cek Plagiasi

Plagiasi_04320016_Fafions Osama Effendy - Fafions Effendy

ORIGINALITY REPORT



		<1 %
11	repository.usd.ac.id Internet Source	<1 %
12	repository.unsri.ac.id Internet Source	<1 %
13	repository.ubb.ac.id Internet Source	<1 %
14	litapdimas.kemenag.go.id Internet Source	<1 %
15	nanopdf.com Internet Source	<1 %
16	smart.stmikplk.ac.id Internet Source	<1 %
17	core.ac.uk Internet Source	<1 %
18	portaldata.org Internet Source	<1 %
19	siat.ung.ac.id Internet Source	<1 %
20	Ayu Tiara Defi, Dedi Suhendro. "RANCANGAN SISTEM INFORMASI PERHITUNGAN PENYUSUTAN FIXED ASSETS MENGGUNAKAN STRAIGHT LINE METHOD PADA PT FIF GROUP PEMATANGSIANTAR", Jurnal Manajemen Informatika dan Sistem Informasi, 2021 Publication	<1 %
21	Feri Prasetyo, Agus Dendi Rachmatsyah, Jumadi Nur, Faisal Adam. "Penerapan Aplikasi	<1 %

Android Pengenalan Huruf Hijaiyah Metode Waterfall Pada Paud Al Fina", Jurnal Sisfokom (Sistem Informasi dan Komputer), 2020

Publication

22	I Nyoman Mahardhika, Ni Putu Tirka Widanti, Ida Ayu Putu Sri Widnyani, Ida Ayu Sayang Bidul. "Evaluasi Kebijakan Pemerintah Terhadap Pengembangan Usaha Mikro, Kecil, dan Menengah (UMKM) di Kabupaten Badung", eCo-Buss, 2024	<1 %
Publication		
23	anythingbutordinary93.blogspot.com	<1 %
	Internet Source	
24	geografi.id	<1 %
	Internet Source	
25	iammuhamadiqbal.wordpress.com	<1 %
	Internet Source	
26	id.scribd.com	<1 %
	Internet Source	
27	repo.darmajaya.ac.id	<1 %
	Internet Source	
28	repositori.umsu.ac.id	<1 %
	Internet Source	
29	repository.dinamika.ac.id	<1 %
	Internet Source	
30	repository.unmuhjember.ac.id	<1 %
	Internet Source	
31	repository.upy.ac.id	<1 %
	Internet Source	

32	www.coursehero.com Internet Source	<1 %
33	www.hashmicro.com Internet Source	<1 %
34	www.tokyo-g-s-theology.com Internet Source	<1 %



Lampiran 2 Poster

