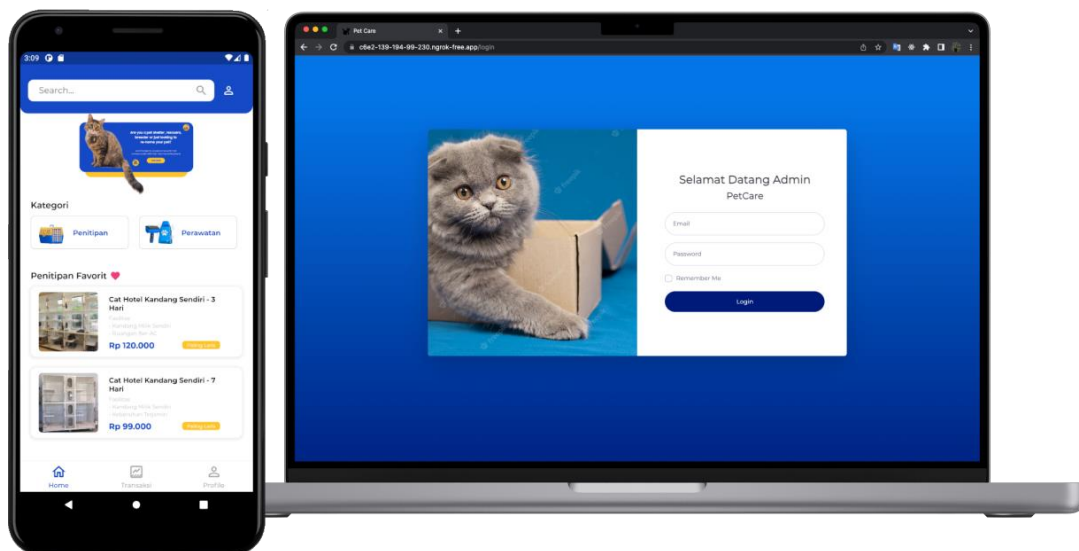


MANUAL SISTEM

"PETPALS+" APLIKASI RESERVASI PERAWATAN DAN PENITIPAN HEWAN BERBASIS ANDROID



Oleh:
Fathur Tias Dewantoro (5200411495)
Anita Fira Waluyo, S.Si., M.Sc.

**UNIVERSITAS TEKNOLOGI YOGYAKARTA
TAHUN 2023**

DAFTAR ISI

DAFTAR ISI	2
DAFTAR GAMBAR	3
TAMPILAN DAN KODE PROGRAM	4
1. Aplikasi Website	4
1.1 Halaman Login Pegawai	4
1.2 Halaman Tabel Layanan.....	5
1.3 Halaman Tambah Layanan.....	7
1.4 Halaman Transaksi.....	8
1.5 Halaman Laporan.....	9
2. Aplikasi Android	11
2.1 Halaman Home Pelanggan.....	11
2.2 Halaman Detail Layanan.....	12
2.3 Halaman Pemesanan	13
2.4 Halaman Daftar Transaksi.....	14
2.5 Halaman Detail Transaksi	15
2.6 Halaman Menu.....	16

DAFTAR GAMBAR

Gambar 1 Halaman Login Pegawai	4
Gambar 2 Potongan Kode Tampilan Login	5
Gambar 3 Potongan Kode Fungsi Login	5
Gambar 4 Halaman Daftar Layanan	6
Gambar 5 Potongan Kode Halaman Daftar Layanan	6
Gambar 6 Potongan Kode Controller Halaman Daftar Layanan	6
Gambar 7 Halaman Tambah Layanan.....	7
Gambar 8 Potongan Kode Tambah Layanan	7
Gambar 9 Potongan Kode Controller Tambah Layanan	8
Gambar 10 Halaman Transaksi Penitipan.....	8
Gambar 11 Kode Tampilan Transaksi Penitipan.....	9
Gambar 12 Kode Controller Transaksi Penitipan	9
Gambar 13 Halaman Pemilihan Jenis Laporan.....	10
Gambar 14 Halaman Cetak Laporan.....	10
Gambar 15 Hasil Laporan Transaksi PDF	10
Gambar 16 Potongan Kode Laporan Penitipan	11
Gambar 17 Halaman Home Pelanggan.....	11
Gambar 18 Potongan Kode Halaman Home Pelanggan	12
Gambar 19 Halaman Detail Layanan.....	12
Gambar 20 Potongan Kode Halaman Detail Layanan	13
Gambar 21 Halaman Pemesanan	13
Gambar 22 Potongan Kode Halaman Pemesanan	14
Gambar 23 Daftar Transaksi Penitipan & Daftar Transaksi Perawatan.....	14
Gambar 24 Potongan Kode List Transaksi	15
Gambar 25 Halaman Detail Transaksi	15
Gambar 26 Potongan Kode Detail Transaksi.....	16
Gambar 27 Halaman Menu.....	16
Gambar 28 Potongan Kode Halaman Menu	17

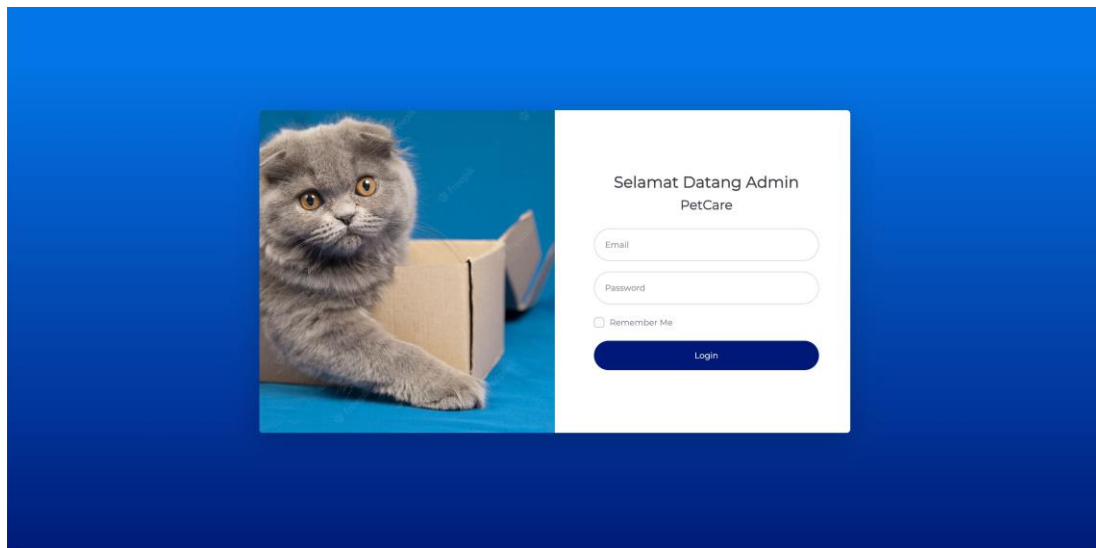
TAMPILAN DAN KODE PROGRAM

Program yang dibangun dibagi menjadi dua platform dengan masing-masing platform memiliki fungsinya sendiri. Platform pertama yang dipilih adalah website, hal ini dipilih karena penggunaan platform website diperuntukkan untuk administrasi dan pengelolaan data. Sehingga pemilihan sistem dengan basis website ini hanya digunakan untuk admin atau pegawai petshop saja. Kemudian aplikasi berbasis android untuk pelanggan melakukan transaksi reservasi. Berikut adalah detail mengenai masing-masing sistem.

1. Aplikasi Website

1.1 Halaman Login Pegawai

Halaman pertama yang akan diakses adalah halaman login dimana halaman ini sebagai portal utama untuk dapat masuk ke dalam sistem dengan memasukkan identitas yang sudah terdaftar di dalam databse. Hanya akun yang memiliki hak akses admin yang dapat masuk ke dalam website, akun dengan hak akses pelanggan hanya dapat masuk ke dalam aplikasi android.



Gambar 1 Halaman Login Pegawai

```

<div class="col-xl-10 col-lg-12 col-md-9 mt-5">
  <div class="card o-hidden border-0 shadow-lg my-5">
    <div class="card-body p-0">
      <!-- Nested Row within Card Body -->
      <div class="row">
        <div class="col-lg-6 d-none d-lg-block bg-login-image"></div>
        <div class="col-lg-6">
          <div class="p-5 mt-5 mb-5">
            <div class="text-center">
              <h1 class="h4 text-gray-900">Selamat Datang Admin/</h1>
              <h1 class="h5 text-gray-900 mb-4">PetCare </h1>
            </div>
            <form class="user" method="POST" action="{{ route('login') }}">
              @csrf
              <div class="form-group">
                <input type="email" class="form-control form-control-user"
                  id="exampleInputEmail" aria-describedby="emailHelp" placeholder="Email"
                  name="email">
              </div>
              <div class="form-group">
                <input type="password" class="form-control form-control-user"
                  id="exampleInputPassword" placeholder="Password" name="password">
              </div>
              <div class="form-group">
                <div class="custom-control custom-checkbox small">
                  <input type="checkbox" class="custom-control-input" id="customCheck">
                  <label class="custom-control-label" form="customCheck">Remember
                    Me/</label>
                </div>
              </div>
              <button class="btn btn-primary btn-user btn-block" type="submit">Login/</button>
            </form>
            @if (session('error'))
              <div class="alert alert-danger">
                <span>Oops!</span> {{ session('error') }}
              </div>
            @endif
          </div>
        </div>
      </div>
    </div>
  </div>

```

Gambar 2 Potongan Kode Tampilan Login

```

public function create()
{
    return view('login');
}

/**
 * Handle an incoming authentication request.
 */
public function store(LoginRequest $request)
{
    $request->authenticate();
    $request->session()->regenerate();

    return redirect()->intended(RouteServiceProvider::HOME);
}

/**
 * Destroy an authenticated session.
 */
public function destroy(Request $request)
{
    Auth::guard('web')->logout();

    $request->session()->invalidate();
    $request->session()->regenerateToken();

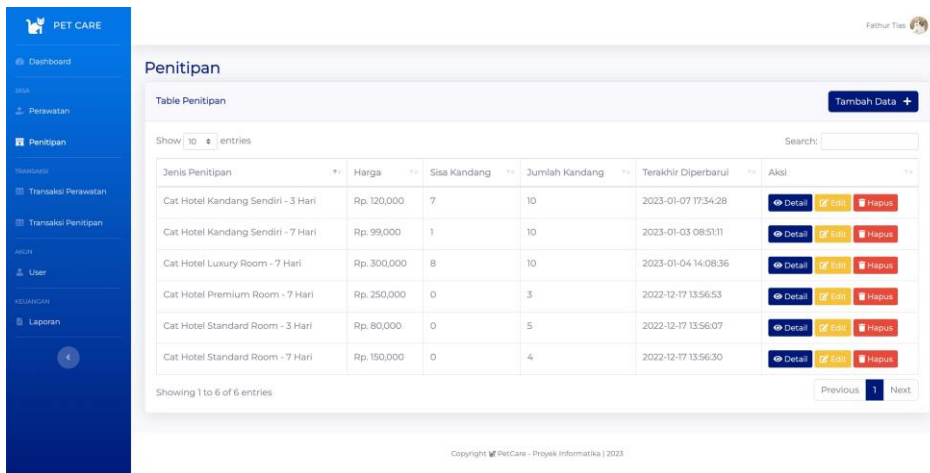
    return redirect('/');
}

```

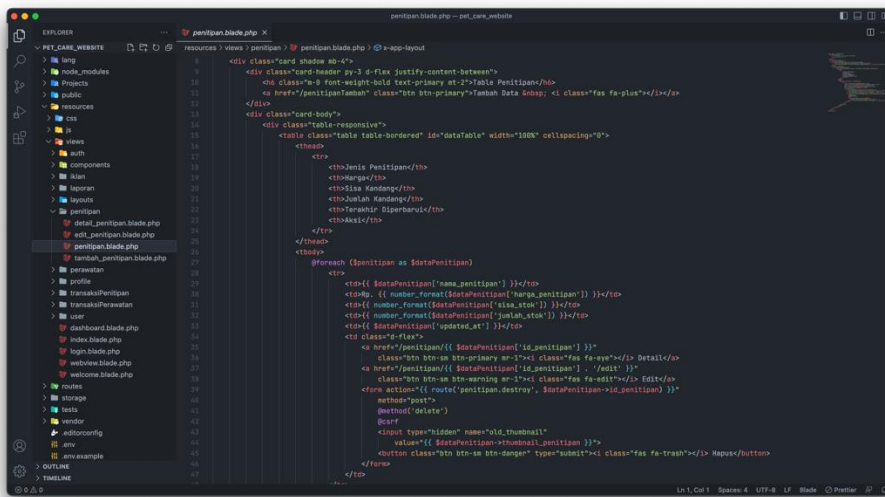
Gambar 3 Potongan Kode Fungsi Login

1.2 Halaman Tabel Layanan

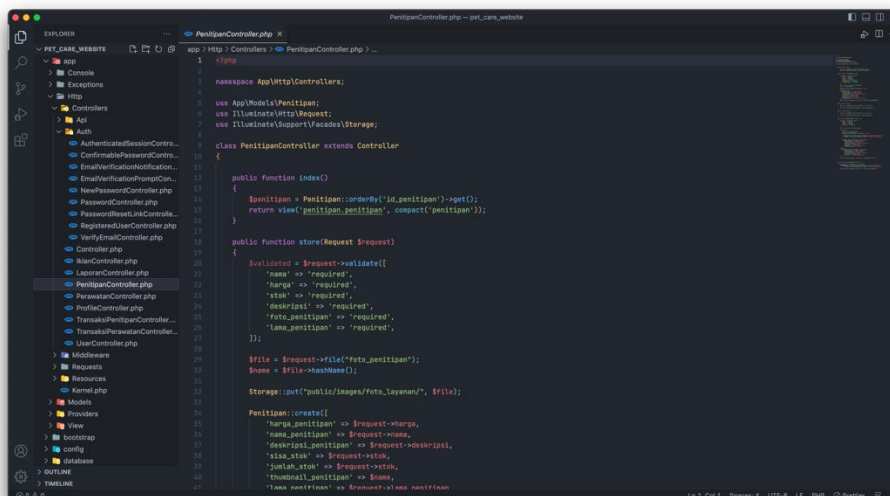
Halaman tabel layanan menampilkan informasi mengenai data layanan yang telah terdaftar. Terdapat beberapa tombol aksi yang dapat dilakukan seperti menambah, mengedit, atau menghapus data layanan.



Gambar 4 Halaman Daftar Layanan



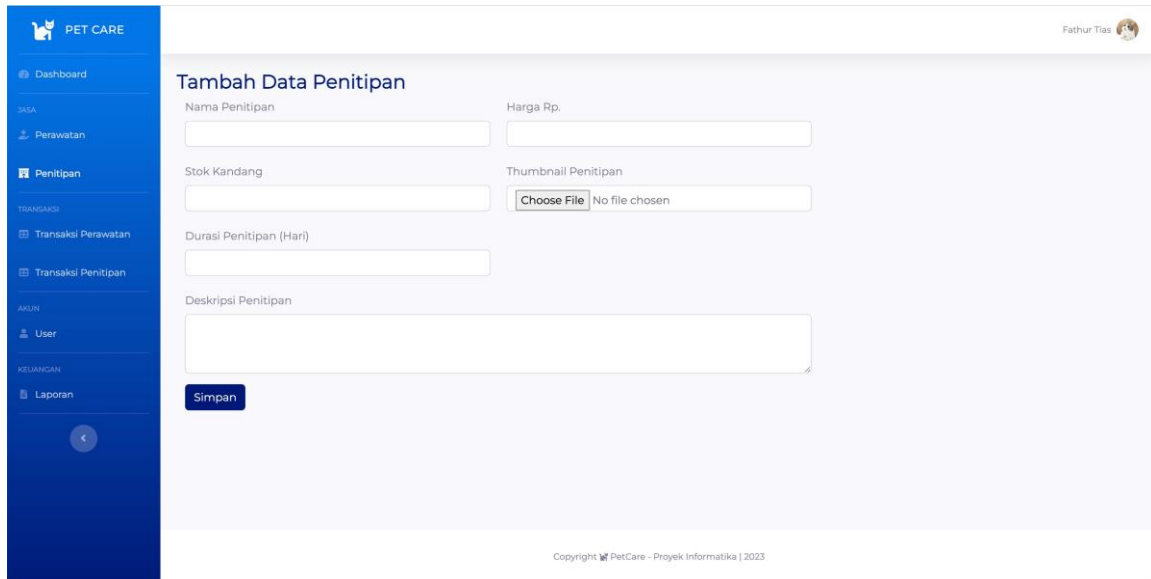
Gambar 5 Potongan Kode Halaman Daftar Layanan



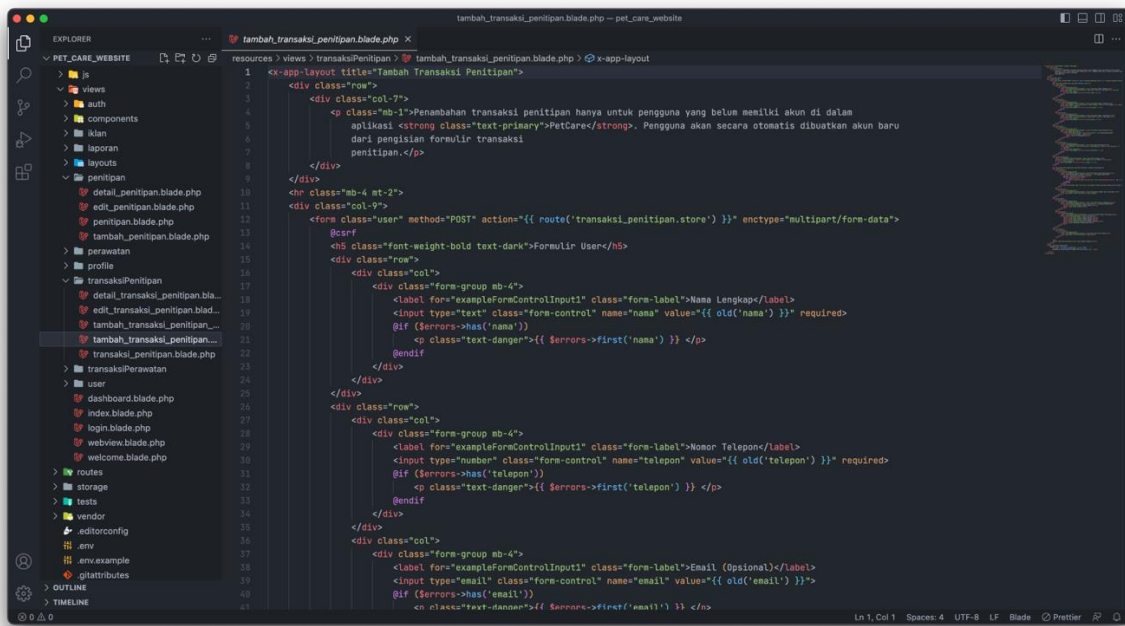
Gambar 6 Potongan Kode Controller Halaman Daftar Layanan

1.3 Halaman Tambah Layanan

Halaman tambah data layanan merupakan halaman yang memperlihatkan formulir untuk menambah layanan baru. Pegawai dapat melakukan penambahan dengan menginputkan informasi yang diminta dari formulir, layanan ini kemudian akan ditampilkan kepada pelanggan melalui aplikasi android.



Gambar 7 Halaman Tambah Layanan



Gambar 8 Potongan Kode Tambah Layanan

```

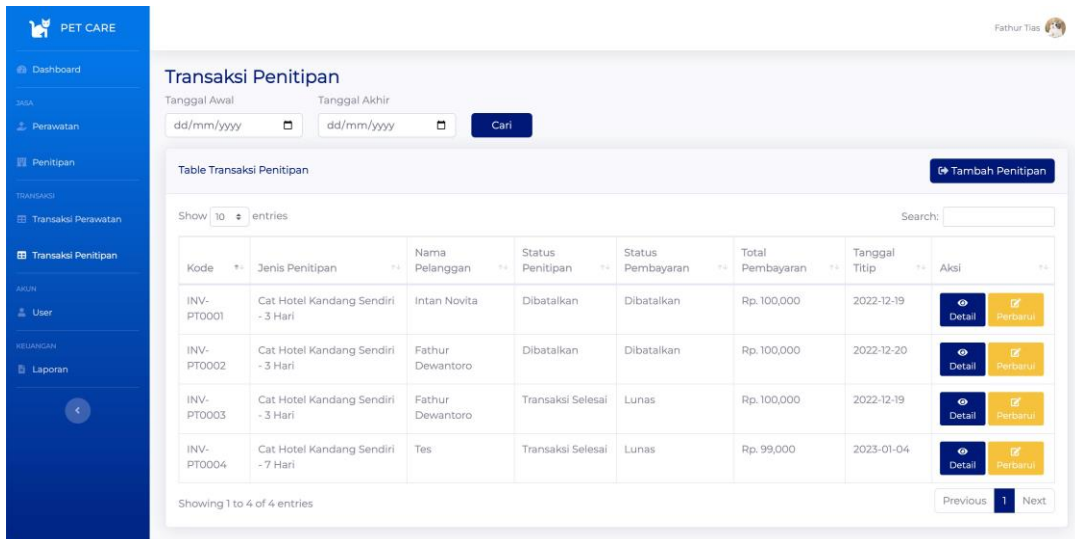
16  }
17
18  public function store(Request $request)
19  {
20      $validated = $request->validate([
21          'nama' => 'required',
22          'harga' => 'required',
23          'stok' => 'required',
24          'deskripsi' => 'required',
25          'foto_penitipan' => 'required',
26          'lama_penitipan' => 'required',
27      ]);
28
29      $file = $request->file('foto_penitipan');
30      $name = $file->hashName();
31
32      Storage::put("public/images/foto_layanan/", $file);
33
34      Penitipan::create([
35          'harga_penitipan' => $request->harga,
36          'nama_penitipan' => $request->nama,
37          'deskripsi_penitipan' => $request->deskripsi,
38          'lama_stok' => $request->stok,
39          'jumlah_stok' => $request->stok,
40          'thumbnail_penitipan' => $name,
41          'lama_penitipan' => $request->lama_penitipan
42      ]);
43      return redirect('/penitipan')->with('info', "Data Berhasil Ditambahkan !");
44  }
45
46  public function show($id)
47  {
48      $penitipan = Penitipan::where('id_penitipan', $id)->first();
49
50      return view('penitipan_detail_penitipan', compact('penitipan'));
51  }
52
53  public function edit($id)
54  {
55      $penitipan = Penitipan::where('id_penitipan', $id)->first();
56  }

```

Gambar 9 Potongan Kode Controller Tambah Layanan

1.4 Halaman Transaksi

Halaman transaksi menampilkan informasi mengenai transaksi yang telah terjadi dan dipesan oleh pelanggan. Pegawai dapat melihat detail mengenai transaksi dan berhak untuk memperbaiki transaksi ketika status transaksi telah berubah. Pegawai dapat melakukan pencarian dan melakukan filter transaksi berdasarkan tanggal transaksi dengan menginputkan tanggal awal dan tanggal akhir sebagai variabel data yang akan ditampilkan.



Gambar 10 Halaman Transaksi Penitipan


```

</div>
<div class="card-body">
  <div class="table-responsive">
    <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
      <thead>
        <tr>
          <th>Kode</th>
          <th>Jenis Penitipan</th>
          <th>Nama Pelanggan</th>
          <th>Status Penitipan</th>
          <th>Status Pembayaran</th>
          <th>Total Pembayaran</th>
          <th>Tanggal Titip</th>
          <th>Aksi</th>
        </tr>
      </thead>
      <tbody>
        @foreach ($transaksiPenitipan as $dataTransaksi)
          <tr>
            <td>{{ $dataTransaksi['id'] }}</td>
            <td>{{ $dataTransaksi['penitipan']->nama_penitipan }}</td>
            <td>{{ $dataTransaksi['user']->name }}</td>
            <td>{{ $dataTransaksi['status_penitipan'] }}</td>
            <td>{{ $dataTransaksi['status_pembayaran'] }}</td>
            <td>Rp. {{ number_format($dataTransaksi['total_pembayaran']) }}</td>
            <td>{{ $dataTransaksi['tanggal_titip'] }}</td>
            <td class="d-flex">
              <a href="/transaksi_penitipan/{{ $dataTransaksi->id }}"
                class="btn btn-sm btn-primary mr-1"><i class="fas fa-eye"></i> Detail</a>
              <a href="/transaksi_penitipan/{{ $dataTransaksi->id }}/edit"
                class="btn btn-sm btn-warning mr-1"><i class="fas fa-edit"></i> Perbarui</a>
            </td>
          </tr>
        @endforeach
      </tbody>
    </table>
  </div>
</div>

```

Gambar 11 Kode Tampilan Transaksi Penitipan

```

namespace App\Http\Controllers;

use App\Models\Penitipan;
use App\Models\TransaksiPenitipan;
use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Generator;
use Illuminate\Support\Facades\Storage;

class TransaksiPenitipanController extends Controller
{
    public function index()
    {
        //return "Transaksi";
        $transaksiPenitipan = TransaksiPenitipan::get();

        return view('transaksiPenitipan.transaksi_penitipan', compact('transaksiPenitipan'));
    }

    public function indexByDate(Request $request)
    {
        $transaksiPenitipan = TransaksiPenitipan::whereBetween('tanggal_titip', [$request->tanggalawal, $request->tanggalakhir])->get();

        return view('transaksiPenitipan.transaksi_penitipan', compact('transaksiPenitipan'));
    }

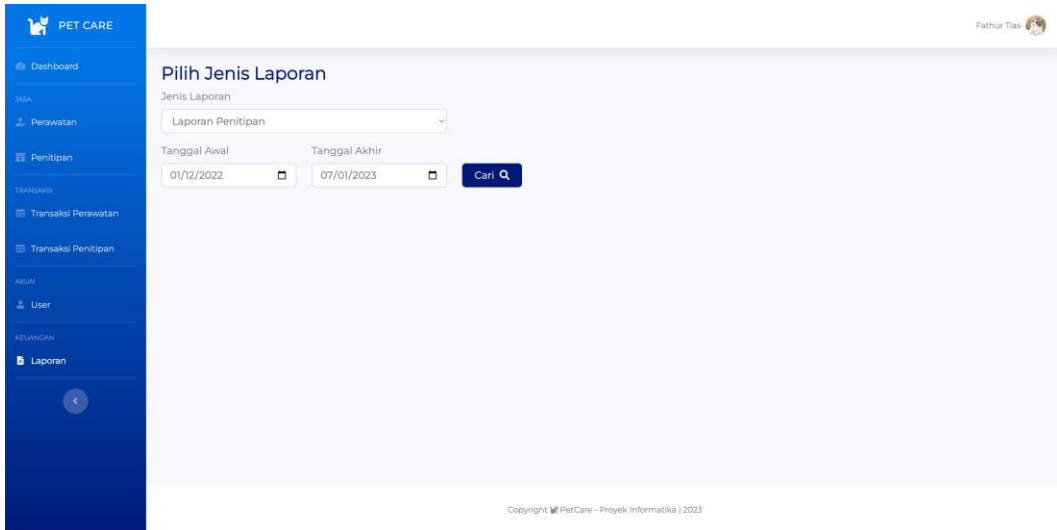
    public function store(Request $request)
    {
        $validated = $request->validate([
            'nama' => 'required',
            'telepon' => 'required',
            'email' => 'required|unique:users,email',
            'password' => 'required',
            'alamat' => 'required',
            'id_penitipan' => 'required',
            'tanggal' => 'required',
            'status_pembayaran' => 'required',
            'status_penitipan' => 'required',
        ]);
    }
}

```

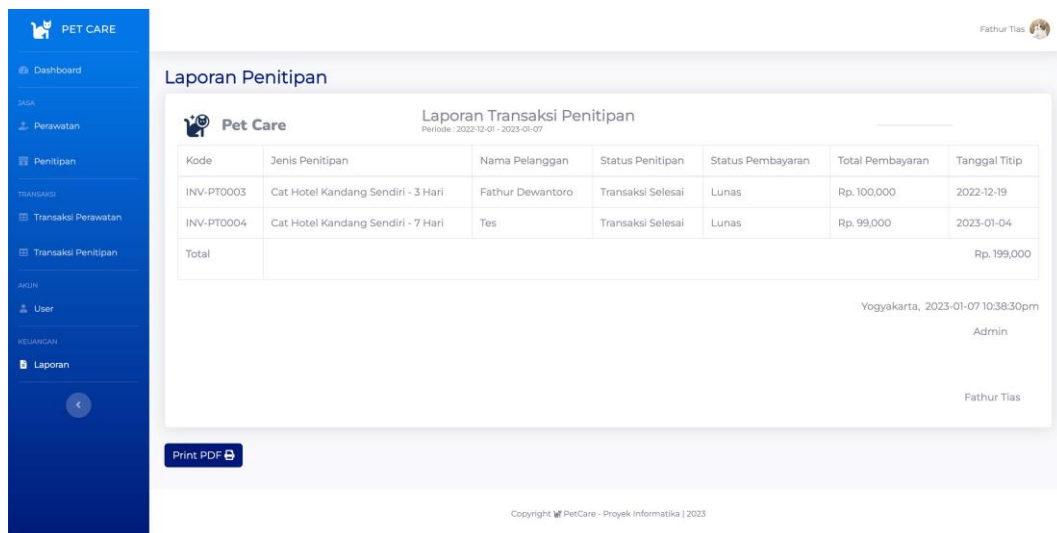
Gambar 12 Kode Controller Transaksi Penitipan

1.5 Halaman Laporan

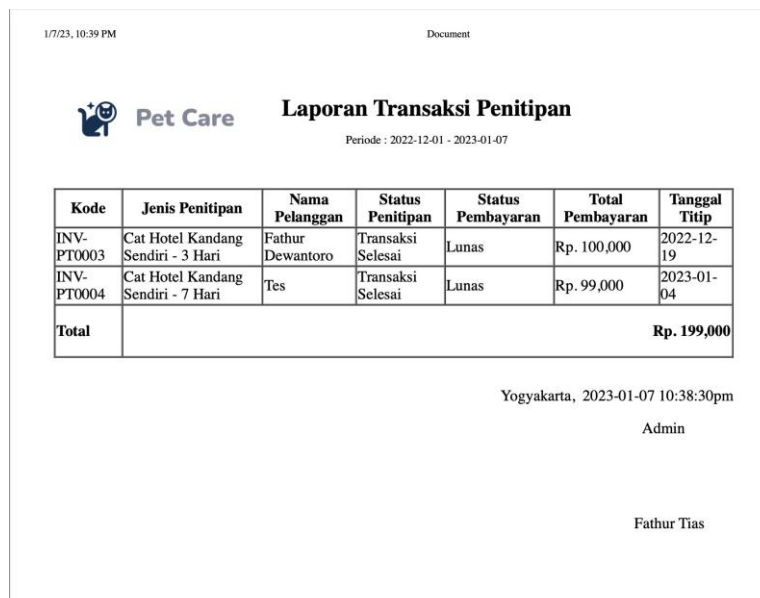
Halaman laporan merupakan tampilan untuk membuat laporan transaksi yang telah terjadi di sehingga dapat dihitung rekapitulasi dari periode tertentu. Laporan dapat di kustomisasi sesuai dengan periode waktu tertentu atau semua transaksi layanan. Terdapat 2 jenis laporan yaitu laporan transaksi penitipan dan laporan transaksi perawatan.



Gambar 13 Halaman Pemilihan Jenis Laporan



Gambar 14 Halaman Cetak Laporan



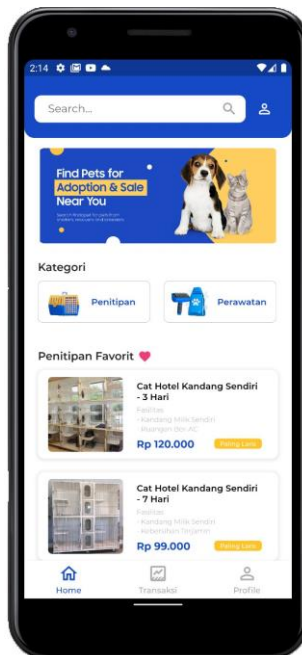
```
resources > views > laporan > table_laporan_penitipan.blade.php > ...
34 <th>Tanggal Titip</th>
35 <th>Total Pembayaran</th>
36 </tr>
37 </thead>
38 <tbody>
39 @foreach ($transaksiPenitipan as $dataTransaksi)
40 @php
41     $total += $dataTransaksi['total_pembayaran'];
42 @endphp
43 <tr>
44 <td>{{ $dataTransaksi['id'] }}</td>
45 <td>{{ $dataTransaksi->penitipan->nama_penitipan }}</td>
46 <td>{{ $dataTransaksi->user->nama }}</td>
47 <td>{{ $dataTransaksi['status_penitipan'] }}</td>
48 <td>{{ $dataTransaksi['status_pembayaran'] }}</td>
49 <td>{{ $dataTransaksi['tanggal_titip'] }}</td>
50 <td>Rp. {{ number_format($dataTransaksi['total_pembayaran']) }}</td>
51 </tr>
52 @endforeach
53 </tbody>
54 </table>
55 </tfoot>
56 </tfoot>
57 <tr>
58 <td style="font-weight: 700; font-size: 16px;">Total</td>
59 <td style="text-align: end; colspan=6">
60 <td style="font-weight: 700; font-size: 16px;">Rp. {{ number_format($total) }} </td>
61 </tr>
62 </tfoot>
63 </table>
64 <div style="text-align: right; margin-top: 30px; margin-right: 50px;">{{ date('Y-m-d H:i:s') }} </div>
65 <div style="text-align: right; margin-right: 50px;">Admin</div>
66 </div>
67 </div>
68 <div style="text-align: right; margin-top: 80px; margin-right: 30px;">{{ Auth::user()->nama }}</div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
```

Gambar 16 Potongan Kode Laporan Penitipan

2. Aplikasi Android

2.1 Halaman Home Pelanggan

Halaman home pelanggan merupakan tampilan pertama kali muncul saat aplikasi android untuk pelanggan dibuka. Pada halaman home tersedia informasi mengenai jenis layanan yang tersedia dan layanan yang terbaru. Terdapat inputan pencarian layanan pada application bar.



Gambar 17 Halaman Home Pelanggan

```
@override
Widget build(BuildContext context) {
  return ListView(
    physics: BouncingScrollPhysics(),
    padding: EdgeInsets.only(top: 0),
    shrinkWrap: true,
    scrollDirection: Axis.vertical,
    children: [
      FutureBuilder(
        future: getKlan(),
        builder: (context, snapshot) {
          if (snapshot.connectionState == ConnectionState.waiting) {
            return Container(
              margin:
                const EdgeInsets.symmetric(horizontal: 20, vertical: 15),
              height: 140,
              decoration: const BoxDecoration(
                color: Colors.grey,
                borderRadius: BorderRadius.all(
                  Radius.circular(10),
                ), // BorderRadius.all
              ), // BoxDecoration
            ); // Container
          } else {
            return CarouselSlider.builder(
              itemCount: dataKlan.length,
              itemBuilder: (context, index, realIndex) {
                return Container(
                  decoration: BoxDecoration(
                    image: DecorationImage(
                      image: NetworkImage(apiUrl +
                        '/storage/images/foto_iklan/' +
                        dataKlan[index].foto), // NetworkImage
                    ), // DecorationImage
                    borderRadius: BorderRadius.all(
                      Radius.circular(10),
                    ), // BorderRadius.all
                  ), // BoxDecoration
                ); // Container
              }, // CarouselBuilder
            );
          }
        },
      ),
    ],
  );
}
```

Gambar 18 Potongan Kode Halaman Home Pelanggan

2.2 Halaman Detail Layanan

Halaman detail layanan merupakan tampilan yang menampilkan informasi mendetail dari layanan yang dipilih pada halaman daftar layanan. Semua informasi mengenai deskripsi, harga, dan ketersediaan stok akan ditampilkan pada halaman ini. Halaman detail juga memiliki sebuah tombol aksi untuk memesan layanan yang ditampilkan.



Gambar 19 Halaman Detail Layanan

```

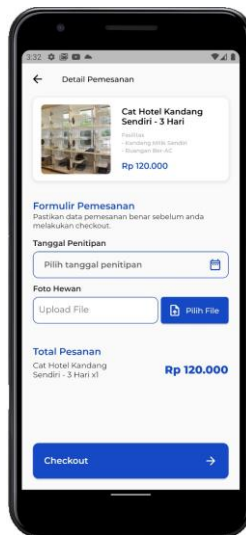
@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.white,
    appBar: AppBar(
      elevation: 0,
      title: const Text(
        'Detail Jasa Penitipan',
        style: TextStyle(
          color: Colors.black, fontWeight: FontWeight.w600, fontSize: 17), // TextStyle
      ), // Text
      backgroundColor: Colors.white,
      leading: IconButton(
        icon: Icon(Icons.arrow_back, color: Colors.black),
        onPressed: () => Navigator.of(context).pop(),
      ), // IconButton
    ), // AppBar
    body: FutureBuilder(
      future: getPenitipan(),
      builder: (BuildContext context, snapshot) {
        if (snapshot.connectionState == ConnectionState.waiting) {
          return const Center(
            child: Text("Loading..."),
          ); // Center
        } else {
          return ListView(
            padding: const EdgeInsets.only(left: 20, right: 20),
            children: [
              Container(
                height: MediaQuery.of(context).size.width * 0.9,
                decoration: BoxDecoration(
                  borderRadius: BorderRadius.circular(10),
                  image: DecorationImage(
                    image: NetworkImage(
                      '$apiUrl/storage/images/foto_layanan/${dataPenitipan.first.thumbnail}'), // NetworkImage
                ), // DecorationImage
              ), // BoxDecoration
            ], // Container
          ); // Container
        }
      },
    ), // FutureBuilder
  ); // Scaffold
}

```

Gambar 20 Potongan Kode Halaman Detail Layanan

2.3 Halaman Pemesanan

Halaman pemesanan merupakan tampilan yang digunakan untuk melakukan pemesanan di tahap akhir. Untuk mengakses halaman pemesanan, pelanggan diharuskan untuk melakukan login terlebih dahulu. Pada halaman pemesanan pelanggan juga diminta untuk menginputkan tanggal transaksi dan foto hewan sebagai identitas hewan.



Gambar 21 Halaman Pemesanan

```

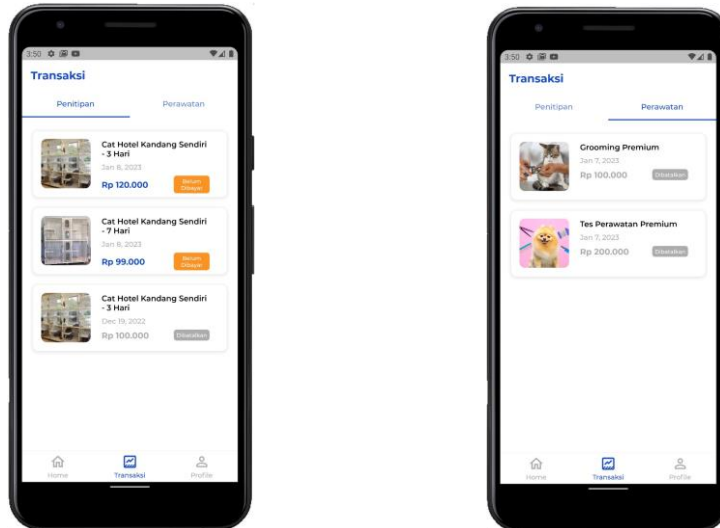
@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey[50],
    appBar: AppBar(
      elevation: 0,
      title: const Text(
        'Detail Pemesanan',
        style: TextStyle(
          color: Colors.black, fontWeight: FontWeight.w500, fontSize: 14), // TextStyle
      ), // Text
      backgroundColor: Colors.grey[50],
      leading: IconButton(
        icon: Icons.arrow_back, color: Colors.black,
        onPressed: () => Navigator.of(context).pop(),
      ), // IconButton
    ), // AppBar
    body: ListView(
      physics: BouncingScrollPhysics(),
      padding: EdgeInsets.only(left: 20, right: 20, top: 20),
      children: [
        const Text(
          'Jasa Dipilih',
          style: TextStyle(
            color: Colors.black,
            type: 'double', // FontWeight.w500
          ), // Text
        ), // Text
      ], // Children
    ), // ListView
  ); // Scaffold
}

```

Gambar 22 Potongan Kode Halaman Pemesanan

2.4 Halaman Daftar Transaksi

Halaman daftar transaksi merupakan tampilan semua daftar transaksi yang telah dilakukan oleh pelanggan. Untuk menampilkan daftar transaksi pelanggan harus melakukan login terlebih dahulu karena daftar transaksi yang muncul berdasarkan identitas akun pelanggan yang melakukan login. Terdapat 2 menu di dalam daftar transaksi yaitu daftar transaksi penitipan dan daftar transaksi perawatan yang dipisah menggunakan tab bar menu.



Gambar 23 Daftar Transaksi Penitipan & Daftar Transaksi Perawatan

```
@override
Widget build(BuildContext context) {
  if (widget.isLogin == true) {
    return DefaultTabController(
      length: 2,
      child: Scaffold(
        backgroundColor: Colors.white,
        appBar: AppBar(
          elevation: 0,
          backgroundColor: Colors.white,
          title: const Text(
            "Transaksi",
            style: TextStyle(
              color: primaryColor,
              fontWeight: FontWeight.w700,
              fontSize: 20,
            ), // TextStyle
          ), // Text
          bottom: const TabBar(
            tabs: [
              Tab(
                text: "Penitipan",
              ), // Tab
              Tab(
                text: "Perawatan",
              ), // Tab
            ],
            indicatorColor: primaryColor,
            labelColor: primaryColor,
          ), // TabBar
          // AppBar
        ), // AppBar
        body: TabBarView(
          children: [
            Padding(
              padding: const EdgeInsets.only(left: 15, right: 15),
              child: FutureBuilder(
                future: getTransaksiPenitipan(),
                builder: (context, snapshot) {
```

Gambar 24 Potongan Kode List Transaksi

2.5 Halaman Detail Transaksi

Halaman detail transaksi merupakan halaman yang menampilkan informasi mendetail mengenai transaksi dari halaman daftar transaksi. Semua informasi seperti kode transaksi, jenis layanan, tanggal layanan, status pembayaran dan status transaksi akan ditampilkan pada halaman ini. Terdapat tombol aksi untuk membatalkan transaksi pada halaman ini.



Gambar 25 Halaman Detail Transaksi

```
class _DetailTransaksiPenitipanState extends State<DetailTransaksiPenitipan> {
  final SharedStorage _sharedStorage = SharedStorage();
  void batalTransaksi() async {
    var token = await _sharedStorage.getStringValuesASF();
    try {
      var response = await http.post(
        Uri.parse(apiUrl + "/api/batalTransaksiPenitipan"),
        headers: <String, String>{
          'Content-Type': 'application/json',
          'Accept': 'application/json',
          'Authorization': 'Bearer $token',
        },
        body: jsonEncode(<String, String>{
          'id_transaksi': widget.dataTransaksi.first.id,
          'id_penitipan': widget.dataTransaksi.first.idPenitipan.toString(),
        }));
      Map<String, dynamic> data = jsonDecode(response.body);
      if (response.statusCode == 200) {
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
          content: Row(
            children: [
              SizedBox(
                width: 10,
                height: 10,
                child: Icon(Icons.check, color: Colors.white),
              ),
              Text("Transaksi berhasil dibatalkan!"),
            ],
          type: SnackBarType.success,
          duration: Duration(seconds: 2),
          backgroundColor: unselectedColor,
          padding: EdgeInsets.only(top: 25, bottom: 25, left: 20),
          behavior: SnackBarBehavior.floating,
          shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(20)),
        ));
      }
    } catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(SnackBar(
        content: Text("Gagal"),
        type: SnackBarType.error,
        duration: Duration(seconds: 2),
        backgroundColor: unselectedColor,
        padding: EdgeInsets.only(top: 25, bottom: 25, left: 20),
        behavior: SnackBarBehavior.floating,
        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(20)),
      ));
    }
  }
}
```

Gambar 26 Potongan Kode Detail Transaksi

2.6 Halaman Menu

Halaman menu merupakan tampilan menu tambahan pada aplikasi seperti menu untuk memperbarui identitas akun dan melakukan logout pada aplikasi.



Gambar 27 Halaman Menu

The image shows a screenshot of an IDE (likely VS Code) with a dark theme. The left sidebar shows a file explorer with a project structure. The main editor area displays the code for a Dart class named `_ProfilePageState`. The code includes state management, authentication logic, and a logout function.

```
21 }
22
23 class _ProfilePageState extends State<ProfilePage> {
24   final SharedStorage _sharedStorage = SharedStorage();
25
26   List<UserModel> _dataUser = [];
27
28   @override
29   void initState() {
30     super.initState();
31     checkLogin();
32   }
33
34   void checkLogin() async {
35     var getToken = await _sharedStorage.isToken();
36     var isUser = await _sharedStorage.isUser();
37     if (getToken) {
38       setState(() {
39         widget.isLogin = getToken;
40         print("Is user ${isUser}");
41       });
42     }
43   }
44
45   Future logout() async {
46     var token = await _sharedStorage.getStringValuesSFC();
47     try {
48       var response =
49         await http.get(Uri.parse('http://192.168.1.100:8080/api/logout'), headers: {
50           'Content-Type': 'application/json',
51           'Accept': 'application/json',
52           'Authorization': 'Bearer ${token}',
53         });
54     } catch (e) {
55       print(e);
56     }
57   }
58 }
59
60
61
```

Gambar 28 Potongan Kode Halaman Menu