

# LAMPIRAN

## Lampiran 1 Surat Izin Penelitian



**UMKT**  
Program Studi  
**Teknik Informatika**  
Fakultas Sains dan Teknologi

Telp. 0541-748511 Fax. 0541-766832  
Website <http://informatika.umkt.ac.id>  
email: [informatika@umkt.ac.id](mailto:informatika@umkt.ac.id)



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Nomor : 026-001/FST.1/A.7/C/2024  
Lampiran : -  
Perihal : Permohonan Izin Penelitian

**Kepada Yth.**  
**Dinas Perpustakaan dan Kearsipan Kota Samarinda**  
di -  
**Tempat**

*Assalamu'alaikum Warrahmatullahi Wabarrakatuh*

Puji Syukur kepada Allah Subhanahu wa ta'ala yang senantiasa melimpahkan Rahmat-Nya kepada kita sekalian. Aamiin.

Sehubungan dengan penyusunan Tugas Akhir/Skripsi Mahasiswa Program Studi Teknik Informatika Tahun Akademik 2023/2024, maka dengan ini kami mengajukan permohonan untuk dapat menerima mahasiswa dari Program Studi Teknik Informatika untuk melakukan penelitian di Dinas Perpustakaan dan Kearsipan Kota Samarinda, dengan nama mahasiswa:

Nama : Muhammad Ramadhani Saputra  
NIM : 2011102441087  
Program Studi : Teknik Informatika

Demikian permohonan ini dibuat. Atas perhatian dan kerjasamanya kami mengucapkan terima kasih.

*Wassalamu'alaikum Warrahmatullahi Wabarrakatuh*

Samarinda, 11 Sya'ban 1445 H  
21 Februari 2024 M



Dua Program Studi S1 Teknik Informatika

*[Handwritten Signature]*  
**Ansyah, S.Kom., M.TI**  
IDN. 1118019203

Kampus 1 : Jl. Ir. H. Juanda, No.15, Samarinda  
Kampus 2 : Jl. Pelita, Pesona Mahakam, Samarinda

## Lampiran 2 Surat Balasan Izin Penelitian



PEMERINTAH KOTA SAMARINDA  
**DINAS PERPUSTAKAAN DAN KEARSIPAN**

Jl. Kesuma Bangsa 5, Samarinda, Kalimantan Timur  
website: <https://perpustakaankearsipan.samarindakota.go.id> --- e-mail: [dispursip.smr@gmail.com](mailto:dispursip.smr@gmail.com)

Samarinda, 23 Juli 2024

Nomor : 400.14.5.4/0895/100.16  
Sifat : Biasa  
Lampiran : -  
Hal : Keterangan Telah Melaksanakan Penelitian

Yth. Ketua Program Studi S-1 Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Muhammadiyah Kalimantan Timur  
di  
Samarinda

Sebagai tindak lanjut surat dari Universitas Muhammadiyah Kalimantan Timur, nomor: 026-001/FST.1/A.7/C/2024, tanggal 21 Februari 2024 tentang Permohonan Izin Penelitian, berkenaan dengan hal tersebut pada prinsipnya kami menerima Mahasiswa di bawah ini untuk melakukan penelitian dan pengambilan data sebagai bahan penyelesaian skripsi.

nama : Muhammad Ramadhani Saputra  
NIM : 2011102441087  
program studi : Teknik Informatika.

Demikian disampaikan, atas kerja sama yang baik diucapkan terima kasih.













Kepala Dinas Perpustakaan  
dan Kearsipan Kota Samarinda,

Erham Yusuf, M.Pd.  
Pembina Utama Muda IV/c  
NIP 196808181998021004


### Lampiran 3 Kartu Kendali Bimbingan

#### KARTU KENDALI BIMBINGAN LAPORAN KARYA ILMIAH


Nama : Muhammad Ramadhani Saputra  
 NIM : 2011102441087  
 Nama Dosen Pembimbing : Abdul Rahim, S.Kom., M.Cs  
 Judul Penelitian : Pengembangan Aplikasi Perpustakaan Berbasis Android Menggunakan Model Waterfall Pada Dinas Perpustakaan Kota Samarinda

No	Tanggal	Uraian Pembimbingan	Paraf Dosen
1.	22/02/2024	Bimbingan Judul Skripsi	
2.	03/03/2024	Bimbingan Kelompok RTA	
3.	14/03/2024	Bimbingan canvas	
4.	24/04/2024	Bimbingan BAB 1	
5.	26/04/2024	Bimbingan bab 2	
6.	28/04/2024	Bimbingan revisi bab 1 dan bab 2	
7.	16/05/2024	Bimbingan untuk Upload proposal ke Simpel	
8.	28/05/2024	Bimbingan revisi review desk	
9.	7/06/2024	Bimbingan bab 3	
10.	17/06/2024	Bimbingan revisi bab 3	
11.			
12.			
13.			
14.			

Dosen Pembimbing

  
 (Abdul Rahim, S.Kom., M.Cs)



  
 (Abdul Rahim, S.Kom., M. TI)

#### Lampiran 4 Jadwal Penelitian

No	Jenis Penelitian	Bulan/2024					
		Feb	Mar	Apr	Mei	Jun	Jul
<b>Tahap Pra Penelitian</b>							
1	Menentukan Judul Penelitian						
2	Menentukan Permasalahan						
3	Mencari Studi Literatur						
4	Menyusun Metode Penelitian						
5	Menentukan Studi Kasus						
6	Menyusun Proposal Penelitian						
7	Review Desk Simpel						
<b>Tahap Penelitian</b>							
1	Analisis Kebutuhan (Requirement Analysis)						
2	Perancangan (Desain)						
3	Implementasi (Implementation)						
4	Pengujian (Testing)						
5	Pemeliharaan (Maintenance)						
<b>Tahap Akhir Penelitian</b>							
1	Penyusunan Laporan						
2	Seminar Hasil						

## Lampiran 5 Code main.dart

```
import 'package:aplikasi_daftar_angota_perpus/screens/form/form.dart';
import 'package:flutter/material.dart';
import 'splashscreens.dart';
import 'screens/login/login.dart';
import 'package:flutter/services.dart';

void main() {
  WidgetsFlutterBinding.ensureInitialized();
  SystemChrome.setPreferredOrientations([
    DeviceOrientation.portraitUp,
  ]);
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      // theme: ThemeData(primaryColor: Colors.green),
      debugShowCheckedModeBanner: false,
      home: Loading(),
      routes: {
        '/form': (context) => MyFormPage(),
        '/login': (context) => LoginPage(),
        '/loading': (context) => Loading(),
      },
    );
  }
}
```

## Lampiran 6 splashscreens.dart

```
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'screens/login/login.dart';
import 'package:percent_indicator/percent_indicator.dart';

class Loading extends StatefulWidget {
  // const Loading({Key? key}) : super(key: key);

  @override
  _LoadingState createState() => _LoadingState();
}
```

```

class _LoadingState extends State<Loading> {
  @override
  void initState() {
    super.initState();
    Timer(const Duration(seconds: 1), () {
      Navigator.of(context)
        .pushReplacement(MaterialPageRoute(builder: (_) => LoginPage()));
    });
  }

  @override
  Widget build(BuildContext context) {
    final MediaQuerywidth = MediaQuery.of(context).size.width;
    return Scaffold(
      backgroundColor:
        Colors.transparent, // Make the scaffold background transparent
      body: Container(
        width: double.infinity,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Colors.white, Color(0xffff012ac0)],
            begin: Alignment.topRight,
            end: Alignment.bottomLeft),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Container(
              ),
            Image.asset(
              'assets/splash_screens/registermember.png', // Replace with your image asset path
              height: 100, // Adjust the height of the image
            ),
          ],
        ),
      ),
    );
  }
}

```

## Lampiran 7 Code login.dart

```
import 'package:aplikasi_daftar_angota_perpus/bottomnavbar/bottomnavbar.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:quickalert/quickalert.dart';
import 'package:shared_preferences/shared_preferences.dart';
import './register.dart';
import 'package:http/http.dart' as http;
// import '../Services/globals.dart';
import '../Services/snackbar.dart';
import 'package:aplikasi_daftar_angota_perpus/Services/auth_services.dart';
import 'dart:convert';
// import 'package:shared_preferences/shared_preferences.dart';

class LoginPage extends StatefulWidget {
  @override
  State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  String _username = "";
  String _password = "";

  loading() {
    QuickAlert.show(
      context: context,
      type: QuickAlertType.loading,
      text: 'Loading...!',
    );
  }

  error(String text) {
    QuickAlert.show(
      context: context,
      type: QuickAlertType.error,
      title: 'Oops...',
      titleColor: Colors.red,
      text: text,
    );
  }

  warning(String text) {
    QuickAlert.show(
      context: context,
      type: QuickAlertType.warning,
      titleColor: Colors.yellow.shade900,
      text: text,
    );
  }
}
```

```

}

loginPressed(BuildContext context) async {
  if (_username.isNotEmpty && _password.isNotEmpty) {
    // Menampilkan loading
    // loading();
    showDialog(
      context: context,
      barrierDismissible: false,
      builder: (BuildContext context) {
        return Center(
          child: SizedBox(
            width: 30,
            height: 30,
            child: CircularProgressIndicator(
              // color: Color(0xffff012ac0),
              color: Color(0xffff012ac0),
            ),
          ),
        );
      },
    );
    await Future.delayed(Duration(milliseconds: 500));
    http.Response response = await AuthServices.login(_username, _password);

    // Menutup dialog loading
    Navigator.pop(context);

    if (response.statusCode == 200) {
      Map<String, dynamic>? responseMap = jsonDecode(response.body);

      if (responseMap != null && responseMap.containsKey('user')) {
        SharedPreferences preferences = await SharedPreferences.getInstance();
        preferences.setInt('id', responseMap['user']['id']);
        preferences.setString('username', responseMap['user']['username']);
        preferences.setString('password', responseMap['user']['password']);
        preferences.setString('firstname', responseMap['user']['firstname']);
        preferences.setString('lastname', responseMap['user']['lastname']);
        preferences.setString('token', responseMap['token']);

        // Navigasi ke halaman Home
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(
            builder: (BuildContext context) => BottomNavbar(),
          ),
        );
      } else {
        errorSnackBar(context,

```



```

        'Respons dari backend tidak sesuai atau tidak mengandung objek user.');
```

```

    }
  } else {
    error("Username atau Password Salah !!");
    // errorSnackBar(context, 'Username atau Password Salah !!');
  }
} else {
  warning('Username dan Password tidak boleh kosong');
  // errorSnackBar(context, 'Masukkan semua kolom yang diperlukan');
}
}
}

```

## Lampiran 8 Code Register.dart

```

import 'dart:convert';
import 'package:aplikasi_daftar_angota_perpus/Services/auth_services.dart';
import 'package:aplikasi_daftar_angota_perpus/screens/login/login.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:http/http.dart' as http;
// import '../Services/globals.dart';
import '../Services/snackbar.dart';

class RegisterPage extends StatefulWidget {
  @override
  State<RegisterPage> createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
  String _firstname = "";
  String _lastname = "";
  String _username = "";
  String _password = "";
  bool _isLoading = false;

  createAccountPressed(BuildContext context) async {
    bool nameValid = RegExp(r'^[a-zA-Z0-9]{3,16}$').hasMatch(_username);
    if (nameValid) {
      showDialog(
        context: context,
        barrierDismissible: false,
        builder: (BuildContext context) {
          return Center(
            child: SizedBox(
              width: 30,
              height: 30,
              child: CircularProgressIndicator(

```

```

        color: Color(0xffff012ac0),
      ),
    ),
  );
},
);

await Future.delayed(Duration(milliseconds: 1500));
http.Response response = await AuthServices.register(
  _username, _password, _firstname, _lastname);
Navigator.pop(context);
Map<String, dynamic> responseMap =
  jsonDecode(response.body) as Map<String, dynamic>;
if (response.statusCode == 200) {
  // Tampilkan snackbar setelah registrasi berhasil
  successSnackBar(context, 'Register Berhasil');
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (BuildContext context) => LoginPage(),
    ),
  );
} else {
  String errorMessage = extractErrorMessage(responseMap);
  errorSnackBar(context, errorMessage);
}
} else {
  errorSnackBar(context, 'Kolom Tidak Boleh Kosong');
}
}
}

```

### Lampiran 9 globals.dart

```

String baseUrl = "http://10.20.2.241:8000/api/";
String baseUrlgambar = "http://10.20.2.241:8000/";
Map<String, String> headers = {
  "Content-Type": "application/json",
  'Accept': 'application/json',
};

```

### Lampiran 10 code form.dart

```

import 'package:aplikasi_daftar_angota_perpus/Services/snackbar.dart';
import 'package:aplikasi_daftar_angota_perpus/loading/loadingform/loadingform.dart';
import 'package:aplikasi_daftar_angota_perpus/screens/form/controller/confirmdata.dart';
import 'package:aplikasi_daftar_angota_perpus/widget/widgetform.dart';
import 'package:aplikasi_daftar_angota_perpus/widget/widgettext.dart';

```

```

import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:shared_preferences/shared_preferences.dart';

// import 'package:http/http.dart' as http;

class MyFormPage extends StatefulWidget {
  @override
  _MyFormPageState createState() => _MyFormPageState();
}

final GlobalKey<ScaffoldMessengerState> scaffoldMessengerKey =
  GlobalKey<ScaffoldMessengerState>();

int userId = 0;

class _MyFormPageState extends State<MyFormPage> {
  void initState() {
    super.initState();
    // Panggil metode untuk mengambil data dari SharedPreferences saat widget diinisialisasi
    getLoginData();
    _delayedPageLoad();
  }

  bool _isLoading = false;
  late String userName;
  late String userToken;

  final formKey = GlobalKey<FormState>();
  final TextEditingController _namaController = TextEditingController();
  final TextEditingController _nikController = TextEditingController();
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _alamatKtp = TextEditingController();
  final TextEditingController _alamatSekarang = TextEditingController();
  final TextEditingController _namaSekolah = TextEditingController();
  final TextEditingController _nomorHp = TextEditingController();
  final TextEditingController _tglLahir = TextEditingController();
  String _listProfesi = 'Siswa';
  String? _gender;
  String _pendidikanTerakhir = 'SD';
  List<String> _schoolTypes = [
    'PAUD',
    'TK',
    'KB',
    'SD',
    'MI',
    'SMP',
    'MTS',

```

```

'SMA',
'SMK',
'MA',
'D3',
'S1',
'S2',
'S3',
];
List<String> _profesi = [
'Siswa',
'Mahasiswa',
'Guru',
'Dosen',
'Peneliti',
'PNS Daerah (Pusat)',
'Karyawan',
'Ti',
'Umum',
'Ibu Rumah Tangga',
];
Future<void> getLoginData() async {
  SharedPreferences preferences = await SharedPreferences.getInstance();
  // Ambil nilai ID, name, dan token dari SharedPreferences
  userId = preferences.getInt('id') ?? 0;
  userName = preferences.getString('name') ?? 'Nama Tidak Ditemukan';
  userToken = preferences.getString('token') ?? "";
  // Panggil setState untuk memperbarui tampilan
  if (mounted) {
    setState(() {});
  }
}

bool isPageLoaded = false;

void _delayedPageLoad() {
  Future.delayed(Duration(milliseconds: 1000), () {
    setState(() {
      isPageLoaded = true; // Setelah 1 detik, halaman dianggap sudah ter-load
    });
  });
}
}

```

## Lampiran 11 profil.dart

```
import 'dart:io';
import 'package:aplikasi_daftar_angota_perpus/bottomnavbar/dialogexit.dart';
import 'package:aplikasi_daftar_angota_perpus/loading/loadingprofil/loadingprofil.dart';
import 'package:aplikasi_daftar_angota_perpus/screens/home.dart';
import 'package:aplikasi_daftar_angota_perpus/widget/widgetprofil.dart';
import 'package:aplikasi_daftar_angota_perpus/widget/widgettext.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import '../Services/auth_services.dart';
import 'dart:convert';
import 'package:shared_preferences/shared_preferences.dart';
import 'dart:async';

class Profil extends StatefulWidget {
  const Profil({Key? key}) : super(key: key);

  @override
  State<Profil> createState() => _ProfilState();
}

late int userId;
late Map<String, dynamic> anggotaData =
  {}; // Menggunakan Map untuk satu anggota

class _ProfilState extends State<Profil> {
  void initState() {
    super.initState();
    // Panggil metode untuk mengambil data dari SharedPreferences saat widget diinisialisasi
    getUserData();
    _delayedPageLoad();
    // Panggil fetchData untuk mengambil data berdasarkan userId
    // fetchDataByIdFlutterUser(1);
  }

  Future<void> getUserData() async {
    SharedPreferences preferences = await SharedPreferences.getInstance();
    // Ambil nilai name, token, dan id dari SharedPreferences
    // userName = preferences.getString('username') ?? 'Nama Tidak Ditemukan';
    userId = preferences.getInt('id') ??
      -1; // Ganti -1 dengan nilai default yang sesuai
    // Panggil setState untuk memperbarui tampilan
    setState(() {});

    // Panggil fetchData dengan userId yang sudah didapat
    fetchDataByIdFlutterUser(userId);
  }
}
```

```

Future<void> fetchDataByIdFlutterUser(int idFlutterUser) async {
  try {
    final response = await API().getDataByIdFlutterUser(idFlutterUser);

    if (response.statusCode == 200) {
      final decodedData = json.decode(response.body);
      if (decodedData is Map<String, dynamic>) {
        // Tidak perlu mengonversi anggota menjadi List
        setState(() {
          anggotaData = decodedData['data'];
          // isLoading = false;
        });
      } else {
        throw Exception('Format respons tidak sesuai');
      }
    } else {
      throw Exception('Gagal mengambil data dari server');
    }
  } catch (error) {
    print('Error: $error');
  }
}

```

## Lampiran 12 screenbuku.dart

```

import 'dart:io';
import 'package:aplikasi_daftar_angota_perpus/bottomnavbar/dialogexit.dart';
import 'package:aplikasi_daftar_angota_perpus/loading/loadingprofil/loadingprofil.dart';
import 'package:aplikasi_daftar_angota_perpus/screens/home.dart';
import 'package:aplikasi_daftar_angota_perpus/widget/widgetprofil.dart';
import 'package:aplikasi_daftar_angota_perpus/widget/widgettext.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import '../Services/auth_services.dart';
import 'dart:convert';
import 'package:shared_preferences/shared_preferences.dart';
import 'dart:async';

class Profil extends StatefulWidget {
  const Profil({Key? key}) : super(key: key);

  @override
  State<Profil> createState() => _ProfilState();
}

late int userId;
late Map<String, dynamic> anggotaData =

```

```

    }); // Menggunakan Map untuk satu anggota

class _ProfilState extends State<Profil> {
  void initState() {
    super.initState();
    // Panggil metode untuk mengambil data dari SharedPreferences saat widget diinisialisasi
    getUserData();
    _delayedPageLoad();
    // Panggil fetchData untuk mengambil data berdasarkan userId
    // fetchDataByIdFlutterUser(1);
  }

  Future<void> getUserData() async {
    SharedPreferences preferences = await SharedPreferences.getInstance();
    // Ambil nilai name, token, dan id dari SharedPreferences
    // userName = preferences.getString('username') ?? 'Nama Tidak Ditemukan';
    userId = preferences.getInt('id') ??
      -1; // Ganti -1 dengan nilai default yang sesuai
    // Panggil setState untuk memperbarui tampilan
    setState(() {});

    // Panggil fetchData dengan userId yang sudah didapat
    fetchDataByIdFlutterUser(userId);
  }

  Future<void> fetchDataByIdFlutterUser(int idFlutterUser) async {
    try {
      final response = await API().getDataByIdFlutterUser(idFlutterUser);

      if (response.statusCode == 200) {
        final decodedData = json.decode(response.body);
        if (decodedData is Map<String, dynamic>) {
          // Tidak perlu mengonversi anggota menjadi List
          setState(() {
            anggotaData = decodedData['data'];
            // isLoading = false;
          });
        } else {
          throw Exception('Format respons tidak sesuai');
        }
      } else {
        throw Exception('Gagal mengambil data dari server');
      }
    } catch (error) {
      print('Error: $error');
    }
  }
}

```

## Lampiran 13 code Login.dart

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login</title>
  <link rel="stylesheet" href="{ {asset('csslogin/logincss.css')}} ">
</head>
<body>
<div class="container" id="container">
  <div class="form-container sign-up-container">
    <form action="#">
      <h1>Create Account</h1>
      <div class="social-container">
        <a href="#" class="social"><i class="fab fa-facebook-f"></i></a>
        <a href="#" class="social"><i class="fab fa-google-plus-g"></i></a>
        <a href="#" class="social"><i class="fab fa-linkedin-in"></i></a>
      </div>
      <span>or use your email for registration</span>
      <input type="text" placeholder="Name" />
      <input type="email" placeholder="Email" />
      <input type="password" placeholder="Password" />
      <button>Sign Up</button>
    </form>
  </div>
  <div class="form-container sign-in-container">
    <form action="/loginproses" method="post">
      @csrf
      <h1>Sign in</h1>
      <br>
      <span>or use your account</span>
      @if($errors->any())
        <div class="alert alert-danger">
          <ul>
            @foreach ($errors->all() as $error)
              <li>{{ $error }}</li>
            @endforeach
          </ul>
        </div>
      @endif
      <input type="text" name="name" placeholder="Username" />
      <input type="password" name="password" placeholder="Password" />
      <br>
      <button>Sign In</button>
    </form>
  </div>
  <div class="overlay-container">
    <div class="overlay">
```



```

<div class="overlay-panel overlay-left">
  <h1>Welcome Back!</h1>
  <p>To keep connected with us please login with your personal info</p>
  <button class="ghost" id="signIn">Sign In</button>
</div>
<div class="overlay-panel overlay-right">
  <h1>Selamat Datang</h1>
  <p>Silakan masukkan username dan password Anda untuk mengakses sistem ini.</p>
</div>
</div>
</div>
</div>
</div>
<script src="{{asset('csslogin/loginjs.js')}}"></script>
</body>
</html>

```

#### Lampiran 14 code dashboard.dart

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">
  <title>Dashboard</title>
  <!-- Custom fonts for this template-->
  <link href="{{asset('sbadmin/all.min.css')}}" rel="stylesheet" type="text/css">
  <link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,700i,800,
800i,900,900i" rel="stylesheet">
  <!-- Custom styles for this template-->
  <link href="{{asset('sbadmin/sb-admin-2.min.css')}}" rel="stylesheet">
  <link href="{{asset('sbadmin/dataTables.bootstrap4.min.css')}}" rel="stylesheet">
</head>
<body id="page-top">
  <!-- Page Wrapper -->
  <div id="wrapper">
    <!-- Sidebar -->
    <ul class="navbar-nav bg-gradient-primary sidebar sidebar-dark accordion" id="accordionSidebar">
      <!-- Sidebar - Brand -->
      <a class="sidebar-brand d-flex align-items-center justify-content-center" href="">
        <div class="sidebar-brand-icon rotate-n-15">
          <i class="">MY</i>
        </div>
        <div class="sidebar-brand-text mx-2">Dashboard</div>
      </a>

```

```

<!-- Divider -->
<hr class="sidebar-divider my-0">
<!-- Nav Item - Dashboard -->
<li class="nav-item active">
  <a class="nav-link" href="/">
    <i class="fas fa-fw fa-tachometer-alt"></i>
    <span>Dashboard</span></a>
  </li>
<hr class="sidebar-divider">
<li class="nav-item">
  <a class="nav-link" href="/daftaranggota">
    <i class="fas fa-fw fa-table"></i>
    <span><strong>Anggota Perpustakaan</strong></span>
  </a>
</li>

<li class="nav-item">
  <a class="nav-link" href="/daftarregister">
    <i class="fas fa-fw fa-table"></i>
    <span><strong>Data Calon Anggota</strong></span></a>
  </li>

<li class="nav-item">
  <a class="nav-link" href="/daftarbuku">
    <i class="fas fa-fw fa-table"></i>
    <span><strong>Buku</strong></span></a>
  </li>

<hr class="sidebar-divider">

```

## Lampiran 15 code anggota perpustakaan

```

@extends('layout.dashboard')
@extends('tambahdata')
@extends('editdata')
@extends('detaildata')
@section('content')
<div class="container-fluid">
  <div class="card shadow mb-4">
    <div class="card-header py-3">
      <h3 class="m-0 font-weight-bold text-primary">
        <center>Data Anggota</center>
      </h3>
    </div>
    <div class="card-body">
      <!-- <a href="#" type="button" class="btn btn-warning mb-2" data-toggle="modal" data-
target="#tambahModal">Tambah +</a -->
      <div class="table-responsive">
        <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">

```

```

<thead>
  <tr>
    <th>no</th>
    <th>Nik</th>
    <th>Nama</th>
    <th>Jenis Kelamin</th>
    <th>No Telpon</th>
    <th>Dibuat</th>
    <th>Aksi</th>
  </tr>
</thead>
<tbody>
  @foreach ($data as $row)
  <tr>
    <th scope="row">{{ $loop->iteration }}</th>
    <td>{{ $row->nik }}</td>
    <td style="text-transform: uppercase;">{{ $row->nama }}</td>
    <td>{{ $row->jeniskelamin }}</td>
    <td>0{{ $row->notelpon }}</td>
    <td>{{ $row->created_at->diffForHumans() }}</td>
    <td>
      <!-- <a href="" type="button" class="btn btn-info" data-toggle="modal" data-
target="#editModal-{{ $row->id }}">Edit</a -->
      <a href="#" type="button" class="btn btn-success" data-toggle="modal" data-
target="#detailModal-{{ $row->id }}">Detail</a>
      <a href="#" type="button" class="btn btn-danger delete" data-id="{{ $row->id }}"
data-nama="{{ $row->nama }}">Delete</a>
      <!-- <a href="/delete/{{ $row -> id }}" type="button" class="btn btn-
danger">Delete</a -->
    </td>
  </tr>
  @endforeach
</tbody>
</table>
</div>
<script src="sbadmin/bootstrap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
<script src="sbadmin/sweetalert.min.js"></script>
<!-- cdn jquery -->
<script src="sbadmin/jquery-3.7.1.slim.min.js" integrity="sha256-
kmHvs0B+OpCW5GVHUNjv9rOmY0IvSIRcf7zGUDTDQM8=" crossorigin="anonymous"></script>
<!-- cdn toastr -->
<script src="sbadmin/toastr.min.js" integrity="sha512-
VEd+nq25CkR676O+plBnDW09R7VQX9Mdiij052gVCp5yVH3jGtH70Ho/UUv4mJDsEdTvqRCFZg0N
KGiojGnUCw=" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
<script>
  $('<span>.delete</span>').click(function() {
    var anggotaid = $(this).attr('data-id')
  })
</script>

```

```

var nama = $(this).attr('data-nama')
swal({
  title: "Yakin ?",
  text: "Kamu akan menghapus data anggota dengan nama *" + nama + "**",
  icon: "warning",
  buttons: true,
  dangerMode: true,
})
.then((willDelete) => {
  if (willDelete) {
    window.location = "/delete/" + anggotaaid + ""
    swal("Data berhasil dihapus", {
      icon: "success",
    });
  } else {
    swal("Data tidak jadi dihapus");
  }
});
})
</script>
</div>
</div>
</div>
@endsection

```

## Lampiran 16 code calon anggota

```

@extends('layout.dashboard')
@extends('tambahdata')

@extends('editdata')
@extends('detaildata')

@section('content')
<div class="container-fluid">
  <div class="card shadow mb-4">
    <div class="card-header py-3">
      <h3 class="m-0 font-weight-bold text-primary">
        <center> Data Calon Anggota</center>
      </h3>
    </div>
    <div class="card-body">
      <!-- <a href="#" type="button" class="btn btn-warning mb-2" data-toggle="modal" data-
target="#tambahModal">Tambah +</a> -->

      <div class="table-responsive">
        <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
          <thead>

```

```

        <tr>
            <th>no</th>
            <th>username</th>
            <th>firstname</th>
            <th>lastname</th>
            <th>Dibuat</th>
            <th>Aksi</th>
        </tr>
    </thead>
    <tbody>
        @foreach ($data as $row)
        <tr>
            <th scope="row">{{ $loop->iteration }}</th>
            <td>{{ $row->username }}</td>
            <td>{{ $row->firstname }}</td>
            <td>{{ $row->lastname }}</td>
            <td>{{ $row->created_at->diffForHumans() }}</td>
            <td>
                <!-- <a href="" type="button" class="btn btn-info" data-toggle="modal" data-
target="#editModal-{{ $row->id }}">Edit</a> -->
                <a href="#" type="button" class="btn btn-danger delete" data-id="{{ $row->id }}"
data-nama="{{ $row->username }}">Delete</a>
                <!-- <a href="/delete/{{ $row -> id }}" type="button" class="btn btn-
danger">Delete</a> -->
                {{-- <a href="#" type="button" class="btn btn-success" data-toggle="modal" data-
target="#detailModal-{{ $row->id }}">Detail</a> --}}
            </td>
        </tr>
        @endforeach
    </tbody>
</table>
</div>
<script src="sbadmin/bootstrap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
<script src="sbadmin/sweetalert.min.js"></script>
<!-- cdn jquery -->
<script src="sbadmin/jquery-3.7.1.slim.min.js" integrity="sha256-
kmHvs0B+OpCW5GVHUNjv9rOmY0IvSIRcf7zGUDTDQM8=" crossorigin="anonymous"></script>
<!-- cdn toastr -->
<script src="sbadmin/toastr.min.js" integrity="sha512-
VEd+nq25CkR676O+pLBnDW09R7VQX9Mdij052gVCP5yVH3jGtH70Ho/UUv4mJDsEdTvqRCFZg0N
KGiojGnUCw==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
<script>
    $('<delete>').click(function() {
        var anggotaid = $(this).attr('data-id')
        var nama = $(this).attr('data-nama')
        swal({
            title: "Yakin ?",

```

```

        text: "Kamu akan menghapus data anggota pegawai dengan nama *" + nama + "*",
        icon: "warning",
        buttons: true,
        dangerMode: true,
    })
    .then((willDelete) => {
        if (willDelete) {
            window.location = "/deleteregister/" + anggotaid + ""
            swal("Data berhasil dihapus", {
                icon: "success",
            });
        } else {
            swal("Data tidak jadi dihapus");
        }
    });
    });
</script>
</div>
</div>
</div>
@endsection

```

### Lampiran 17 code detail data

```

@foreach ($data as $row)
<div class="modal fade" id="detailModal-{{ $row->id }}" tabindex="-1" role="dialog" aria-
labelledby="tambahModalLabel"
    aria-hidden="true">
    <div class="modal-dialog modal-lg" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="tambahModalLabel">Detail Data</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <form action="/updatedata/{{ $row->id }}" method="POST" enctype="multipart/form-data">
                    @csrf

                    <div class="form-row">
                        <div class="form-group col-md-6">
                            <label for="nik">NIK:</label>
                            <div class="input-group">
                                <input type="number" class="form-control" id="nik-{{ $row->nik }}" name="nik"
                                    value="{{ $row->nik }}" readonly>
                                <div class="input-group-append">
                                    <button type="button" class="btn btn-primary copy-button"

```

```

        onclick="copyToClipboard(this)">
        <i class="fas fa-copy"></i>
    </button>
</div>
</div>
</div>

```

### Lampiran 18 code routes/api.php

```

Route::post('/auth/register', [AuthContoller::class, 'register']);
Route::post('/auth/login', [AuthContoller::class, 'login']);
Route::post('/test', function () {
    return response([
        'message' => "Api is working"
    ],200);
});

Route::get('/jumlah-anggota', [AnggotaApiController::class, 'jumlah']);
Route::get('/anggota/{id_flutteruser}', [AnggotaApiController::class, 'getDataByIdFlutterUser']);
Route::post('/anggota/{id_flutteruser}', [AnggotaApiController::class, 'anggota']);
Route::put('/anggota/{id}', [AnggotaApiController::class, 'update']);
Route::get('/daftaranggota/ceknik', [AnggotaApiController::class, 'ceknik']->name('ceknik'));

Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
    return $request->user();
});

Route::get('/daftar-buku', [BukuApiController::class, 'index']);
Route::get('/gambarbuku/{filename}', [BukuApiController::class, 'getGambar']);
Route::get('/daftarbuku/{kategori}', [BukuApiController::class, 'index']);
Route::get('/daftarbukuid/{id}', [BukuApiController::class, 'daftarbukuid']);
Route::get('/search', [BukuApiController::class, 'search']);

```

### Lampiran 19 routes/web.php

```

Route::get('/', function () {
    $jumlahanggota = daftaranggota::count();
    $jumlahanggotacowo = daftaranggota::where('jeniskelamin','Laki-laki')->count();
    $jumlahanggotacewe = daftaranggota::where('jeniskelamin','Perempuan')->count();
    return view('welcome', compact('jumlahanggota','jumlahanggotacowo','jumlahanggotacewe'));
})->middleware('auth');
// Route::get('/login', function () {
//     return view('login.login');
// });

Route::get('/daftaranggota', [DaftaranggotaController::class, 'index']->name('daftaranggota')->middleware('auth');

```

```

Route::get('/daftarregister', [DaftaranggotaController::class, 'daftarregister']->name('daftarregister')->middleware('auth'));
Route::get('/deleteregister/{id}', [DaftaranggotaController::class, 'deleteregister']->name('deleteregister'));
Route::post('/insertdata', [DaftaranggotaController::class, 'insertdata']->name('insertdata'));
Route::post('/updatedata/{id}', [DaftaranggotaController::class, 'updatedata']->name('updatedata'));
Route::get('/delete/{id}', [DaftaranggotaController::class, 'delete']->name('delete'));

Route::get('/login', [LoginController::class, 'login']->name('login'));
Route::post('/loginproses', [LoginController::class, 'loginproses']->name('loginproses'));
Route::get('/register', [LoginController::class, 'register']->name('register'));
Route::post('/registeruser', [LoginController::class, 'registeruser']->name('registeruser'));
Route::get('/logout', [LoginController::class, 'logout']->name('logout'));

Route::get('/daftarbuku', [DaftarbukuController::class, 'daftarbuku']->name('daftarbuku'));
Route::post('/insertdata', [DaftarbukuController::class, 'insertdata']->name('insertdata'));
Route::get('/deletebuku/{id}', [DaftarbukuController::class, 'deletebuku']->name('deletebuku'));

```

## Lampiran 20 code app/http/controllers/logincontroller.php

```

<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Support\Str;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class LoginController extends Controller
{
    public function login(){
        // return view('login.login');
        return view('login.logintest');
    }
    public function loginproses(Request $request){
        if(Auth::attempt($request->only('name', 'password'))){
            return redirect('/');
        }

        return redirect('login')->withErrors(['message' => 'Nama pengguna atau kata sandi salah']);
    }

    public function register(){

```



```

        return view('login.register');
    }
    public function registeruser(Request $request){
        User::create([
            'name' => $request->name,
            'email' => $request->email,
            'password' => bcrypt($request->password),
            'remember_token' => Str::random(60),
        ]);

        return redirect('/login');
        // dd($request->all());
    }
    public function logout(){
        Auth::logout();
        return redirect('login');
    }
}
}

```

#### Lampiran 21 code app/http/controllers/daftaranggotacontroller.php

```

<?php

namespace App\Http\Controllers;

use App\Models\daftaranggota;
use App\Models\FlutterUser;
use Illuminate\Http\Request;

class DaftaranggotaController extends Controller
{
    public function index()
    {
        $data = daftaranggota::all();

        return view('daftaranggota', compact('data'));
    }

    public function insertdata(Request $request)
    {
        daftaranggota::create($request->all());

        return redirect()->route('daftaranggota')->with('success', 'Data Berhasil Ditambah');
    }

    public function updatedata(Request $request, $id)
    {
        $data = daftaranggota::find($id);
    }
}

```

```

        $data->update($request->all());
        return redirect()->route('daftaranggota')->with('success', 'Data Berhasil Diupdate');
    }

    public function delete($id){
        $data = daftaranggota::find($id);
        $data->delete();
        return redirect()->route('daftaranggota')->with('success', 'Data Berhasil Dihapus');
    }

    public function daftarregister() {
        $data = FlutterUser::all();
        return view('daftarregister', compact('data'));
    }

    function deleteregister($id){
        $data = FlutterUser::find($id);
        $data->delete();
        return redirect()->route('daftarregister')->with('success','Data Berhasil Dihapus');
    }
}

```

## Lampiran 22 code app/http/controllers/anggotaapicontroller.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\daftaranggota;
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;

class AnggotaApiController extends Controller
{
    public function index()
    {
        $anggota = daftaranggota::all();
        return response()->json(['message' => 'Success', 'data' => $anggota,]);
    }

    public function anggota(Request $request)
    {
        $request->validate([
            'id_flutteruser' => 'unique:daftaranggotas,id_flutteruser',

```

```

// tambahkan aturan validasi lainnya sesuai kebutuhan
]);
Sanggota = daftaranggota::create($request->all());
return response()->json(['message' => 'success', 'data' => $anggota]);
}

public function update(Request $request, $id)
{
    $anggota = daftaranggota::find($id);
    $anggota->update($request->all());
    return response()->json(['message' => 'success', 'data' => $anggota]);
}

public function getDataByIdFlutterUser(Request $request, $id_flutteruser)
{
    // Menggunakan join untuk mengambil data sesuai id_flutteruser
    $anggota = daftaranggota::join('flutterusers', 'daftaranggotas.id_flutteruser', '=', 'flutterusers.id')
    ->where('daftaranggotas.id_flutteruser', $id_flutteruser)
    ->select('daftaranggotas.*', 'flutterusers.username')
    ->first();

    if ($anggota) {
        return response()->json(['message' => 'Success', 'data' => $anggota]);
    } else {
        return response()->json(['message' => 'Data not found'], 404);
    }
}
}

```

## Lampiran 23 wawancara

No	Pertanyaan	Jawaban
1	<p>Bagaimana proses pendaftaran perpustakaan yang saat ini berlaku di Dinas Perpustakaan Kota Samarinda?</p>	<p>Putry Desintia Ramadhani menjelaskan bahwa proses pendaftaran anggota perpustakaan di Dinas Perpustakaan Kota Samarinda saat ini dilakukan secara manual. Calon anggota harus datang langsung ke perpustakaan, mengisi formulir pendaftaran, dan menyerahkan dokumen yang diperlukan seperti fotokopi KTP atau kartu identitas lainnya. Petugas perpustakaan kemudian memverifikasi data yang diberikan sebelum memberikan kartu anggota.</p>
2	<p>Apa saja kendala atau masalah yang sering dihadapi dalam proses pendaftra anggota perpustakaan?</p>	<p>Menurut Putry Desintia Ramadhani, kendala utama yang sering dihadapi dalam proses pendaftaran anggota perpustakaan adalah kesalahan dalam pengisian formulir pendaftaran oleh calon anggota dan keterlambatan dalam proses verifikasi data oleh petugas. Terdapat beberapa masalah tambahan, seperti ketika kertas formulir habis, petugas perpustakaan harus mencetak kertas formulir terlebih dahulu, yang memakan banyak waktu. Selain itu, terdapat masalah ketika kertas print khusus untuk formulir habis, sehingga petugas harus mencarinya di gudang</p>

- atau membelinya terlebih dahulu untuk mendapatkan kertas tersebut. Hal ini mengakibatkan berkurangnya efektivitas dan efisiensi terhadap layanan perpustakaan.
- 3 Setelah calon anggota mengisi fomulir dan di serahkan ke petugas, apakah fomulir itu di simpan untuk dibackup atau dibuang?  
Putry Desintia Ramadhani menerangkan bahwa formulir pendaftaran yang telah diisi oleh calon anggota tidak disimpan sebagai backup, melainkan dibuang setelah diproses.
  - 4 Fitur-fitur apa saja yang diharapkan dapat dimiliki oleh aplikasi pendaftaran anggota perpustakaan berbasis android ini?  
Putry Desintia Ramadhani mengharapkan aplikasi pendaftaran anggota perpustakaan berbasis Android ini memiliki berbagai fitur yang memudahkan proses pendaftaran dan pengelolaan data anggota. Fitur-fitur yang diharapkan antara lain adalah:
    - Login: Memungkinkan pengguna untuk masuk ke aplikasi dengan akun yang telah terdaftar.
    - Register: Memungkinkan calon anggota untuk mendaftar dan membuat akun baru.
    - Menginput data: Memungkinkan pengguna untuk mengisi dan menyimpan data pendaftaran anggota baru.
    - Menghapus data: Memungkinkan petugas untuk menghapus data anggota yang tidak valid atau tidak diperlukan lagi.
    - Mencari data: Memungkinkan petugas untuk mencari data anggota berdasarkan kriteria tertentu.
    - Melihat detail data: Memungkinkan petugas untuk melihat detail data anggota secara lengkap dan terperinci.
  - 5 Bagaimana sistem verifikasi dan validasi data calon anggota yang saat ini diterapkan di Dinas Perpustakaan Kota Samarinda?  
Putry Desintia Ramadhani menjelaskan bahwa sistem verifikasi dan validasi data calon anggota yang saat ini diterapkan di Dinas Perpustakaan Kota Samarinda dilakukan secara manual oleh petugas perpustakaan. Petugas memeriksa keaslian dokumen yang diserahkan oleh calon anggota dan memastikan bahwa semua informasi yang diberikan sesuai dengan persyaratan.
  - 6 Berapa rata-rata pendaftaran anggota baru di perpustakaan perhari?  
Menurut Putry Desintia Ramadhani, perpustakaan menerima 15 hingga 20 anggota baru setiap harinya.
-

# SKRIPSI MUHAMMAD RAMADHANI SAPUTRA

by Teknik Informatika UMKT



---

**Submission date:** 16-Jul-2024 02:04PM (UTC+0800)

**Submission ID:** 2417648066

**File name:** 087\_Muhammad\_Ramadhani\_Saputra\_-\_MUHAMMAD\_RAMADHANI\_SAPUTRA.docx (6.14M)

**Word count:** 8244

**Character count:** 52500

## SKRIPSI MUHAMMAD RAMADHANI SAPUTRA

### ORIGINALITY REPORT

**22%**

SIMILARITY INDEX

**21%**

INTERNET SOURCES

**7%**

PUBLICATIONS

**7%**

STUDENT PAPERS

### PRIMARY SOURCES

1	<a href="http://widuri.raharja.info">widuri.raharja.info</a> Internet Source	2%
2	<a href="http://repository.ar-raniry.ac.id">repository.ar-raniry.ac.id</a> Internet Source	1%
3	Submitted to Universitas Pamulang Student Paper	1%
4	<a href="http://repository.ub.ac.id">repository.ub.ac.id</a> Internet Source	1%
5	<a href="http://repo.unikadelasalle.ac.id">repo.unikadelasalle.ac.id</a> Internet Source	1%
6	<a href="http://repository.uin-suska.ac.id">repository.uin-suska.ac.id</a> Internet Source	1%
7	<a href="http://ojs.trigunadharma.ac.id">ojs.trigunadharma.ac.id</a> Internet Source	1%
8	<a href="http://repository.its.ac.id">repository.its.ac.id</a> Internet Source	1%
9	<a href="http://123dok.com">123dok.com</a> Internet Source	<1%